



**A TIGHTLY-COUPLED
INS/GPS INTEGRATION
USING A MEMS IMU**

THESIS

Jonathan M. Neu

AFIT/GE/ENG/04-19

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GE/ENG/04-19

A TIGHTLY-COUPLED INS/GPS INTEGRATION USING A
MEMS IMU

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the
Degree of Master of Science

Jonathan M. Neu, BSEE

September 2004

Approved for public release; distribution unlimited

AFIT/GE/ENG/04-19

A TIGHTLY-COUPLED INS/GPS INTEGRATION USING A
MEMS IMU

Jonathan M. Neu, BSEE

Approved:



Dr. John F. Raquet
Thesis Advisor

7 SEP 04
Date



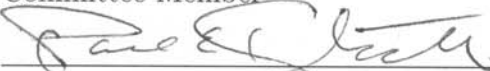
Dr. Peter S. Maybeck
Committee Member

7 Sep 04
Date



Dr. Tom S. Wailes
Committee Member

8 SEP 04
Date



Captain Paul E. Kladitis
Committee Member

7 SEP 04
Date

Acknowledgements

I would like to thank my advisor, Dr. John Raquet, for his expert advice and guidance during my time at AFIT. His flexibility and patience, as well as his technical expertise, were key in overcoming many obstacles. I would also like to thank my committee members for their advice and for their work spent reviewing this thesis, especially Dr. Peter Maybeck for his knowledge of Kalman filters. A special thanks goes to Don Smith for his excellent technical support, and to Jay Schamus, who provided the matrix libraries so essential to the programming.

I want to thank my wife for her love and patience while I worked on my thesis. Without her constant support, this would have been a much more difficult process. Finally, thanks to God for this educational opportunity and for the many blessings He has given me.

Jonathan M. Neu

Table of Contents

	Page
Acknowledgements	iii
List of Figures	vii
List of Tables	x
Abstract	xi
 I. Introduction	 1-1
1.1 Background	1-1
1.1.1 Inertial Navigation Systems	1-1
1.1.2 Global Position System	1-1
1.1.3 Integrated INS/GPS	1-2
1.1.4 MEMS	1-3
1.1.5 Potential Applications	1-4
1.2 Problem Definition	1-5
1.3 Related Research	1-5
1.3.1 Tightly-coupled INS/GPS	1-5
1.3.2 MEMS Sensors in Navigation Applications . .	1-6
1.4 Methodology	1-7
1.5 Thesis Overview	1-8
 II. Background	 2-1
2.1 Overview	2-1
2.2 Navigation Coordinate Systems	2-1
2.3 Inertial Navigation Systems	2-1

	Page
2.3.1 INS Principle of Operation	2-1
2.3.2 INS Implementation	2-3
2.3.3 INS Errors	2-5
2.4 Global Positioning System	2-9
2.4.1 GPS Principle of Operation [23]	2-9
2.4.2 Errors in the Pseudorange Measurement . . .	2-10
2.5 Kalman Filters	2-12
2.5.1 State Model Equations	2-14
2.5.2 Measurement Model Equations	2-15
2.5.3 System Propagate and Update Equations . . .	2-16
2.6 Test Equipment	2-17
2.6.1 XSens MT-9B	2-17
2.6.2 NovAtel Black Diamond System	2-18
2.7 Summary	2-19
III. System Development	3-1
3.1 Overview	3-1
3.2 Algorithm	3-1
3.2.1 Filter states	3-1
3.2.2 System Dynamics Model	3-2
3.2.3 Measurement Model	3-6
3.3 Code Structure	3-9
3.4 Summary	3-12
IV. Testing and Analysis	4-1
4.1 Overview	4-1
4.2 Data Methodology	4-3
4.2.1 Data Collection	4-4

	Page
4.2.2 Data Correction	4-5
4.2.3 Data Processing	4-6
4.3 Filter Features	4-6
4.3.1 System Alignment	4-7
4.3.2 Instrument Error Feedback	4-8
4.3.3 INS Resets	4-11
4.4 Filter Tuning	4-13
4.4.1 Initial State Covariances	4-13
4.4.2 Noise Values	4-15
4.5 Test Results	4-16
4.5.1 Short-term Performance	4-16
4.5.2 Long-term Performance	4-21
4.5.3 GPS Outage	4-33
4.6 Residuals	4-37
4.7 Summary	4-38
V. Conclusions and Recommendations	5-1
5.1 Conclusions	5-1
5.2 Recommendations	5-2
Appendix A. Position Track of Relevant Data Sets	A-1
Bibliography	BIB-1
Vita	VITA-1

List of Figures

Figure		Page
1.1.	Block Diagram of Typical Tightly-Coupled INS/GPS	1-3
2.1.	INS Operation [33]	2-3
2.2.	INS Implementations [33]	2-5
2.3.	A MEMS Accelerometer [2]	2-7
2.4.	A Vibratory MEMS Gyroscope [21]	2-8
2.5.	Satellite Navigation [23]	2-10
2.6.	XSens MT-9B IMU	2-18
2.7.	NovAtel Black Diamond System (INS/GPS)	2-19
3.1.	Filter Measurement Model	3-7
3.2.	Code Flowchart	3-10
4.1.	BDS North and East Position Track, “figure eights”	4-2
4.2.	BDS North and East Position Track, “short-term”	4-2
4.3.	BDS North and East Position Track, “long-term”	4-3
4.4.	System Block Diagram	4-7
4.5.	Filter Estimates of Z-axis Accelerometer and Gyroscope Biases, “long-term”	4-10
4.6.	MT-9 and BDS Indicated Down Velocities, with and without Feedback, “figure eights”	4-11
4.7.	North Position Filter Performance, with and without Feedback, “long-term”	4-12
4.8.	North, East, and Altitude Filter Performance, “short-term” .	4-17
4.9.	North, East, and Down Velocity Filter Performance, “short- term”	4-17
4.10.	Roll and Pitch Filter Performance, “short-term”	4-18

Figure		Page
4.11.	Yaw Filter Performance, “short-term”	4-18
4.12.	Position Errors and Standard Deviations, “short-term”	4-19
4.13.	Velocity Errors and Standard Deviations, “short-term”	4-19
4.14.	X, Y, and Z-Axis Tilt Errors and Standard Deviations, “short-term”	4-20
4.15.	North, East, and Altitude Filter Performance, “long-term” . .	4-22
4.16.	North, East, and Down Velocity Filter Performance, “long-term”	4-23
4.17.	Roll and Pitch Filter Performance, “long-term”	4-23
4.18.	Yaw Filter Performance, “long-term”	4-24
4.19.	Position Errors and Standard Deviations, “long-term”	4-24
4.20.	Velocity Errors and Standard Deviations, “long-term”	4-25
4.21.	X, Y, and Z-Axis Tilt Errors and Standard Deviations, “long-term”	4-25
4.22.	Zoomed-in Yaw Filter Performance, “long-term”	4-27
4.23.	Clock Bias and Clock Bias Drift Estimates, “long-term” . . .	4-29
4.24.	Clock Bias and Clock Bias Drift Standard Deviations, “long-term”	4-30
4.25.	X, Y, and Z Accelerometer Bias Estimates, “long-term” . . .	4-30
4.26.	X, Y, and Z Accelerometer Bias Standard Deviations, “long-term”	4-31
4.27.	X, Y, and Z Gyroscope Bias Estimates, “long-term”	4-31
4.28.	X, Y, and Z Gyroscope Bias Standard Deviations, “long-term”	4-32
4.29.	North and East Filter Performance, GPS Outage	4-34
4.30.	North and East Velocity Filter Performance, GPS Outage . .	4-34
4.31.	North and East Errors and Standard Deviations, GPS Outage	4-35
4.32.	North and East Velocity Errors and Standard Deviations, GPS Outage	4-35
4.33.	Filter Residuals, “short-term”	4-38

Figure		Page
A.1.	BDS North and East Position Track, “figure eights”	A-1
A.2.	BDS Altitude Track, “figure eights”	A-2
A.3.	BDS North, East, and Down Velocities, “figure eights”	A-3
A.4.	BDS Roll, Pitch, and Yaw, “figure eights”	A-4
A.5.	BDS North and East Position Track, “short-term”	A-5
A.6.	BDS Altitude Track, “short-term”	A-6
A.7.	BDS North, East, and Down Velocities, “short-term”	A-7
A.8.	BDS Roll, Pitch, and Yaw, “short-term”	A-8
A.9.	BDS North and East Position Track, “long-term”	A-9
A.10.	BDS Altitude Track, “long-term”	A-10
A.11.	BDS North, East, and Down Velocities, “long-term”	A-11
A.12.	BDS Roll, Pitch, and Yaw, “long-term”	A-12

List of Tables

Table		Page
2.1.	Approximate Magnitudes of Pseudorange Errors [18] [25] [28] [32]	2-12
4.1.	Data Set Summary	4-3
4.2.	Initial Filter State Covariances	4-14
4.3.	System Dynamics Noise (Q matrix)	4-15
4.4.	Short Term Performance Errors	4-20
4.5.	Long Term Performance Errors	4-26
4.6.	GPS Outage Performance Errors	4-36

Abstract

Micro-Electro-Mechanical Systems (MEMS) technology holds great promise for future navigation systems because of the reduced size and cost of MEMS inertial sensors relative to conventional devices. Current MEMS devices are much less accurate than standard inertial sensors, but they can still be useful. In this thesis, data was recorded from an inexpensive MEMS inertial measurement unit and integrated with GPS measurements using a tightly-coupled Kalman filter. The overall goal of this research is to investigate the usefulness of MEMS sensors for a small, real-time, low-cost INS/GPS integration.

A golf cart was used to collect dynamic data, along with a commercial INS/GPS system to provide reference data. This data was then post-processed, and the filter's performance in the position, velocity, and attitude outputs were evaluated by comparing them to the reference system. The important system features of system alignment, bias feedback, and INS resets are described, and the filter's performance is analyzed using its estimate and covariance outputs and comparing them to the true error. Filter residuals are also shown and discussed.

The final results show that, with adequate processing available, the INS/GPS filter using the MEMS instruments provides good position, velocity, and attitude results over a period of up to 15 minutes, as long as the data is at least somewhat dynamic. Without vehicle motion, the vehicle yaw state tends to wander excessively, due to the bias and noise of the MEMS gyroscopes. Over a long static period, the filter's position outputs would most likely diverge and become unstable. Recommendations are made to combat this problem, among them to conduct more characterization of the MEMS sensors, and to add GPS velocity measurements as an input to the filter.

A TIGHTLY-COUPLED INS/GPS INTEGRATION USING A MEMS IMU

I. Introduction

1.1 Background

1.1.1 Inertial Navigation Systems. Navigation technology in the form of an Inertial Navigation System (INS) is a well-known and well-defined technology, having been used in United States military equipment such as ships, submarines, and aircraft since the 1960's [33]. On a basic level, an INS works by measuring acceleration and rotation in all three directions. By knowing its starting location and attitude, it can determine its present location. However, current inertial technology is limited in its accuracy; an INS with a position drift of 1/2 to 1/4 nautical miles/hour is considered quite good for an inertial system, and is typically the standard accuracy level used on a vehicle such as a long-range aircraft [33]. Such a system can also be quite expensive to purchase, on the order of tens of thousands of dollars (see Section 2.6.2). Gimballed systems tend to have higher maintenance costs because of their mechanical complexity (strapdown inertial systems naturally have lower maintenance costs and breakdown rates because of many fewer moving parts).

1.1.2 Global Position System. The advent of the Global Positioning System (GPS) has provided users with an accurate and cheap navigation system (compared to an INS), although one that is vulnerable to jamming or other signal outages. The GPS consists of a number of satellites that orbit the earth, monitored and operated by a network of ground stations. These satellites transmit signals that are received by GPS receivers, which use the signals to determine the receiver's range to the

satellite. If enough satellites are available (four minimum), the receiver performs multilateration to determine its position on (or above) the earth. A basic, affordable GPS receiver (on the order of a hundred dollars) can obtain position accuracies of 10 meters or less on a regular basis [23].

1.1.3 Integrated INS/GPS. By combining a GPS receiver with an INS using a Kalman filter, the more accurate navigation solution of GPS can be obtained while still having the robust capability of an INS, which provides attitude measurements and system functionality in the presence of GPS signal outages. Another benefit of combining the two systems is that each system complements the other's weaknesses very well. GPS has excellent low-frequency performance but poor high-frequency performance; from one second to another the error in the indicated position may change by up to several meters, but over longer periods of time the position error remains relatively constant. An INS has very good high-frequency performance but poor low-frequency performance; over a very short period of time (seconds), the error in indicated position changes very little, but within minutes or hours the position drifts significantly, on the order of kilometers. A combined INS/GPS has the benefits of both systems and displays very good low and high-frequency performance. The Kalman filter is a common way of implementing this integration [20], and there are several different levels of integration [14]. A loosely-coupled system is one that integrates the position solution from both the INS and the GPS receiver. This is the easiest coupling to implement, but it sacrifices some of the advantages of a tighter integration. A tightly-coupled system uses raw pseudorange data from the GPS receiver and acceleration and angular rate measurements from the Inertial Measurement Unit (IMU), the accelerometer-gyroscope triad at the heart of an INS. Combining the two systems in such a tightly-coupled manner provides improved performance, allowing the INS to aid the GPS receiver tracking loops, and allowing the overall solution to use GPS data even when fewer than four satellites are available. A notional block diagram of a tightly-coupled implementation can be seen in Figure

1.1. This figure shows a feedforward implementation, which means that the INS is allowed to run freely once it is initialized. A feedback implementation would periodically provide the INS with filter corrections.

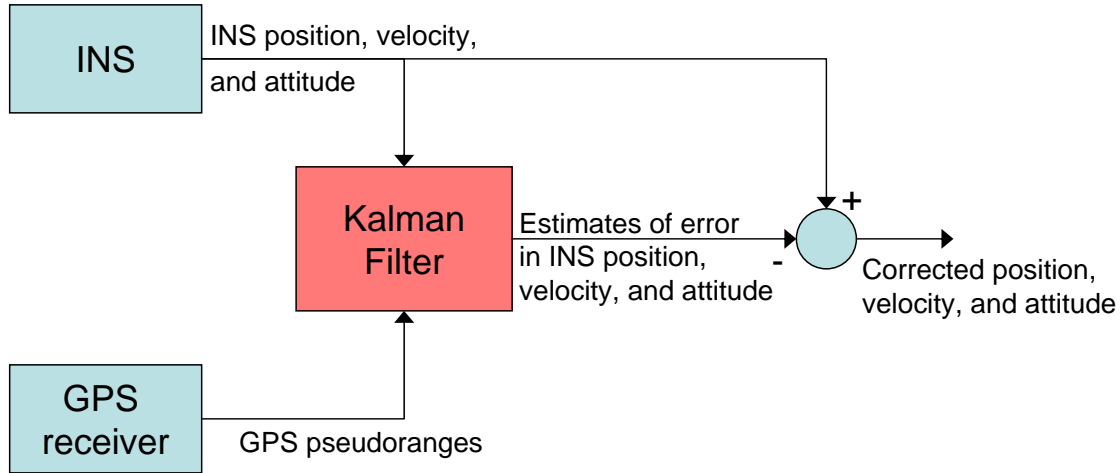


Figure 1.1 Block Diagram of Typical Tightly-Coupled INS/GPS

There is also ultra-tight coupling, which uses the IMU and GPS receiver data in its rawest form [14]. The goal is to wring even more performance out of a combined INS/GPS system, but this ultra-tight technique is still fairly new and not fully developed. It also requires access to the internal GPS receiver tracking loops, which is not always feasible. A combined system, integrated in one of the ways described, is much more accurate than a stand-alone INS, but because of the cost of a standard IMU, its cost is still too prohibitive to be practical for common usage.

1.1.4 MEMS. The advent of Micro-Electro-Mechanical Systems (MEMS) technology has great potential for the navigation field. One big advantage of MEMS technology is low cost. MEMS devices are batch-produced in very large quantities, making the cost per unit very cheap. Another advantage is weight and volume savings. Gyroscopes and accelerometers can be somewhat large and heavy devices, and the MEMS versions of these instruments are orders of magnitude smaller and lighter. MEMS inertial instruments also have much lower power requirements than

their full-size counterparts, making them ideal for mobile applications. The main disadvantage to MEMS devices, at least at this stage in their technological development, is their poor performance when compared to standard inertial instruments. Gyroscope performance tends to be the limiting factor in the accuracy of an INS, and MEMS inertial sensors are no different, with MEMS gyroscopes currently limiting MEMS navigation performance. However, MEMS technology is improving at a rapid rate with much ongoing research. Several MEMS IMUs are currently being developed for military uses (such as the Honeywell HG-1900), and others are already available in the regular commercial market [22] [39].

1.1.5 Potential Applications. An INS/GPS combination that uses a MEMS IMU provides great cost savings that allow such a system to be used in many different applications. One such application is vehicle navigation. GPS systems are already in use for such navigation systems, but as vehicles travel under bridges, near or under natural obstructions like trees, or through “urban canyon” environments, GPS signals drop out or are blocked from view. With a combined INS/GPS system, the INS could continue tracking the vehicle’s position until the GPS signal is reacquired (and it also would provide information about the vehicle’s heading and attitude, if desired). Another possible application is ejection seat testing, for which the trajectory and attitude of an ejection seat need to be tracked and logged. An INS/GPS system would be much more efficient, and much cheaper, than the current camera-based tracking system used by the Air Force [34]. A third application is for tracking the orientation and attitude of large shipboard antennas that are used for precision landing systems for aircraft carriers [26]. A fourth application is for night-vision goggles that have pre-mapped terrain information overlaid on the wearer’s vision, because such a system would need to know the position and attitude of the wearer’s head [4]. In short, any system that could benefit from navigation information would have a low-cost solution available.

1.2 Problem Definition

The long-term objective of this line of research is to develop a real-time, tightly-coupled INS/GPS integration using a MEMS IMU. MEMS is desirable because of its cost and size advantages. A tightly-coupled integration is to be used because of the performance advantage it provides over a loosely-coupled system. Tight coupling is also less computationally intense than ultra-tight coupling—an important consideration in a real-time system. The ideal end result is a component that can easily be integrated into another system, providing position, velocity, and attitude data in a standardized digital format to allow maximum interoperability. Such a system would be unique in the fact that little, if any, other research has analyzed the broad navigation performance of a general-purpose real-time INS/GPS integration that uses a low-quality MEMS IMU as a foundation. Cost, power, and size, as well as accuracy, are all important factors in the development process of such a device.

To that end, this thesis describes the development and testing of the MEMS IMU/GPS Kalman filter integration algorithm, implemented in a post-processing mode. Implementing these algorithms in a real-time system is left for future development.

1.3 Related Research

1.3.1 Tightly-coupled INS/GPS. Vallot, et al [35] contains an excellent example of a Kalman filter INS/GPS integration very similar to the one developed in this research. In demonstrating a Differential INS/GPS system for aircraft landing guidance, they built a tightly-coupled integration that differs in only a few respects. Their approach was to use an error state filter with eighteen states: three position, three velocity, three attitude, one pressure altitude, user clock frequency, user clock phase, three accelerometer biases, and three gyroscope biases. Their Kalman filter processed and updated the error estimates every ten seconds, while using prefiltered GPS data taken at a 1 Hz data rate. This prefiltered GPS data consisted of pseudo-

range and pseudorange rate. The most obvious difference is that Vallot et al's system used a high-quality IMU (with ring-laser gyroscopes), so the poor performance of a MEMS IMU was not an issue.

1.3.2 MEMS Sensors in Navigation Applications. With the improving quality of MEMS inertial sensors, interest is high and much new research is occurring in the area of MEMS INS/GPS integration. One example is Martin, et al [19], who examined the performance of a MEMS INS/GPS integration for the vehicle navigation application mentioned in Section 1.1. Using a low-cost MEMS IMU in their tightly-coupled INS/GPS, they received good results by using the INS/GPS to calibrate the vehicle's odometer and magnetometer attitude reference. Another example is Ford, et al [12], who document the loosely-coupled integration of a GPS receiver and a Honeywell HG-1900 MEMS IMU. Their system was tested in a vehicle on a road course, and their results show the significant benefits an INS provides during GPS signal outage. Van Graas and Farrell [36] developed a tightly-coupled MEMS INS/GPS integration that "differs radically from all other approaches reported up to this time." Their system processes GPS code and carrier measurements separately, and uses double or even triple-differencing in their measurements (depending on the mode of operation). The INS portion of the system focuses only on the dominant error characteristics and uses segmented processing, separating the dynamics and position estimators. Their final results showed that centimeter/second velocity accuracy was feasible while using a low-cost MEMS IMU. There are numerous other examples of MEMS INS applications [7] [8] [9] [29], a notable example being a MEMS Attitude and Heading Reference System (AHRS) in the process of gaining FAA flight certification [38].

Interest is also high in methods to compensate for the comparatively poor performance of MEMS inertial instruments. Hide, et al [17] explored three methods of implementing a Kalman filter to compensate specifically for this. These methods are covariance scaling, the Adaptive Kalman Filter (AKF), and Multiple Model

Adaptive Estimation (MMAE). Covariance scaling applies an artificial scale factor to the predicted covariance in order to weight the measurements. The AKF attempts to make the filter’s residuals “consistent with their theoretical covariances”. The MMAE method uses several filters that run simultaneously, each with different stochastic properties. Their simulations show that it is possible to cut INS dynamic alignment time by 5-10 times by using these methods. Wagner and Kasties [37] experimented with multiple GPS antennas combined with long lever arms between the inertial instruments and the antennas. According to their results, the increased state observability given by the multiple antenna-long lever arm configuration has significant potential to improve MEMS INS/GPS performance. Another unique method was proposed by El-Diasty and El-Rabbany, who used a modular neural network to suppress successfully MEMS sensor high-frequency noise components [11].

The relatively new technology of MEMS holds great promise for the future of navigation. The expectation of the navigation community is that MEMS low-cost navigation will soon begin to approach and even surpass conventional inertial instruments in performance and utility [3] [24].

1.4 Methodology

The first step in the system development was to learn the basics of a tightly-coupled Kalman filter INS/GPS integration. This was accomplished by using and modifying previously existing MATLAB code [27] to test and refine the system algorithm (and also to provide some preliminary filter tuning). The next step was to transfer the algorithm to the C++ programming language, in the hopes that the C++ code could be used in the future real-time system with little modification. Once the code was completed, testing and tuning were conducted as detailed in Chapter 4.

1.5 Thesis Overview

Chapter 2 provides more in-depth background information and appropriate theory about INS, the GPS, and Kalman filter integrations. Chapter 3 presents and develops the Kalman filter dynamics and measurement models, and details the overall software structure. Chapter 4 presents test results, including final tuning parameters and final system performance. Finally, Chapter 5 contains the overall conclusion and suggests the next steps to be taken in this line of study.

II. Background

2.1 Overview

This chapter provides background information and basic theory for all of the systems integrated and studied in this research. First, the relevant reference frames are defined. Next, the basic operation of an inertial navigation system (INS) is discussed, along with a description of the dominant error characteristics. The third section describes the Global Positioning System (GPS), the fourth section describes Kalman filter theory and operation, and the final section describes the MEMS IMU and the truth reference system used in testing.

2.2 Navigation Coordinate Systems

All navigation solutions are expressed in some reference frame, which is merely a way of relating a user's position to the earth. The two frames used in this research are the Latitude-Longitude-Height (LLH) reference frame and the North-East-Down (NED) frame. The LLH frame uses Latitude and Longitude in degrees and height in meters to describe a position relative to the earth. The NED frame is a local-level Cartesian frame that uses a point on the earth as the origin, and uses North, East, and Down as its X, Y, and Z axes respectively (units are in meters).

2.3 Inertial Navigation Systems

The development in this section is taken largely from [33], which is an excellent source of further information about INS theory and operation.

2.3.1 INS Principle of Operation. Inertial navigation is based on a basic law of physics—that a body in motion in a straight line will continue its uniform motion unless an external force acts on the body. This external force, when acting

on the body, produces an acceleration proportional to the force. If this acceleration can be measured, when integrated once (with respect to time) it gives information about the body's change in velocity, and when integrated again, the body's position change is obtained. This principle can be expanded to three dimensions if the angular rotation of the body can be measured. With knowledge of a body's acceleration and rotation in three dimensions, the body's change in position, velocity, and attitude relative to an external reference frame can be calculated. The only requirement is that the initial starting point and orientation relative to the external reference frame (also called the navigation frame) is known.

The basics of an INS algorithm can be seen in Figure 2.1. The rotation sensors (gyroscopes) measure the vehicle's rotation and allow the INS to “know” its orientation with respect to the navigation frame. The accelerometers measure the specific forces (the combined effects of gravity and acceleration) acting on the vehicle. Because the vehicle's orientation is known, the effects of gravity can be removed from the specific force measurements, and what remains is the vehicle's actual acceleration caused by changes in its motion. Two integrations are performed—the first providing the vehicle's velocity, and the second providing position.

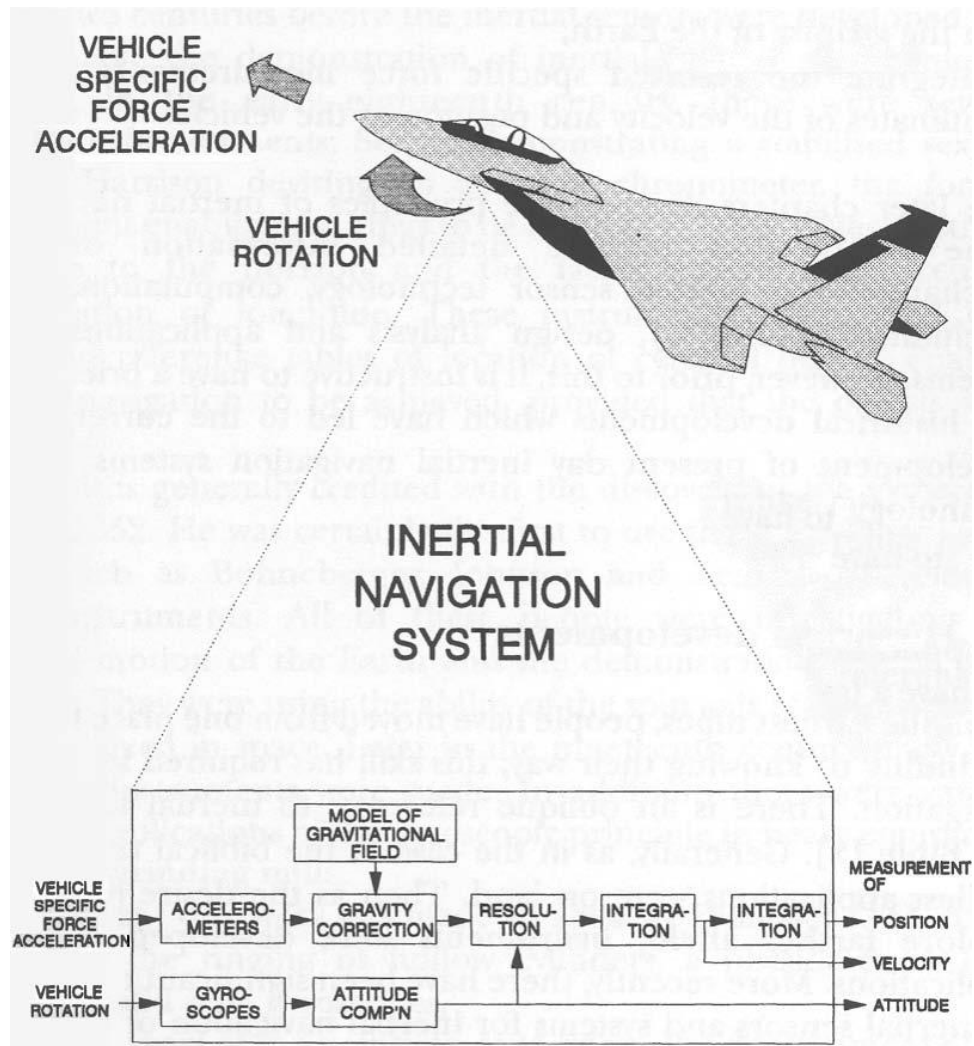


Figure 2.1 INS Operation [33]

2.3.2 INS Implementation. The sensors that provide the INS with its measurements are accelerometers and gyroscopes. Each sensor measures acceleration or rotation in one dimension, so an INS has a triad of each kind of sensor, with the sensitive axes of each sensor usually mounted orthogonally to the others. There are two basic ways to implement an INS: a platform mechanization and a strapdown system. A platform INS relies on its gyroscopes to stabilize a gimballed platform on which the sensors are mounted, and to keep it at a certain attitude with respect to the navigation frame. In this manner, a vehicle can keep track of its orientation

with respect to the navigation frame by referencing the vehicle's orientation with respect to the INS platform. A strapdown system operates in much the same way, but keeps track of the vehicle's orientation with respect to the navigation frame by continuously calculating a direction cosine matrix (DCM), which is a matrix that relates two separate measurement frames. In this case, the DCM relates the vehicle frame to the navigation frame.

Platform INS's were the first to gain widespread use (in the 1960s), with strapdown systems gaining wider use as processing power and gyroscope accuracy increased. Modern INS's tend to use a strapdown implementation, although for the most accurate systems, platform mechanizations are still used (see Figure 2.2 [33]). This research uses a strapdown implementation.

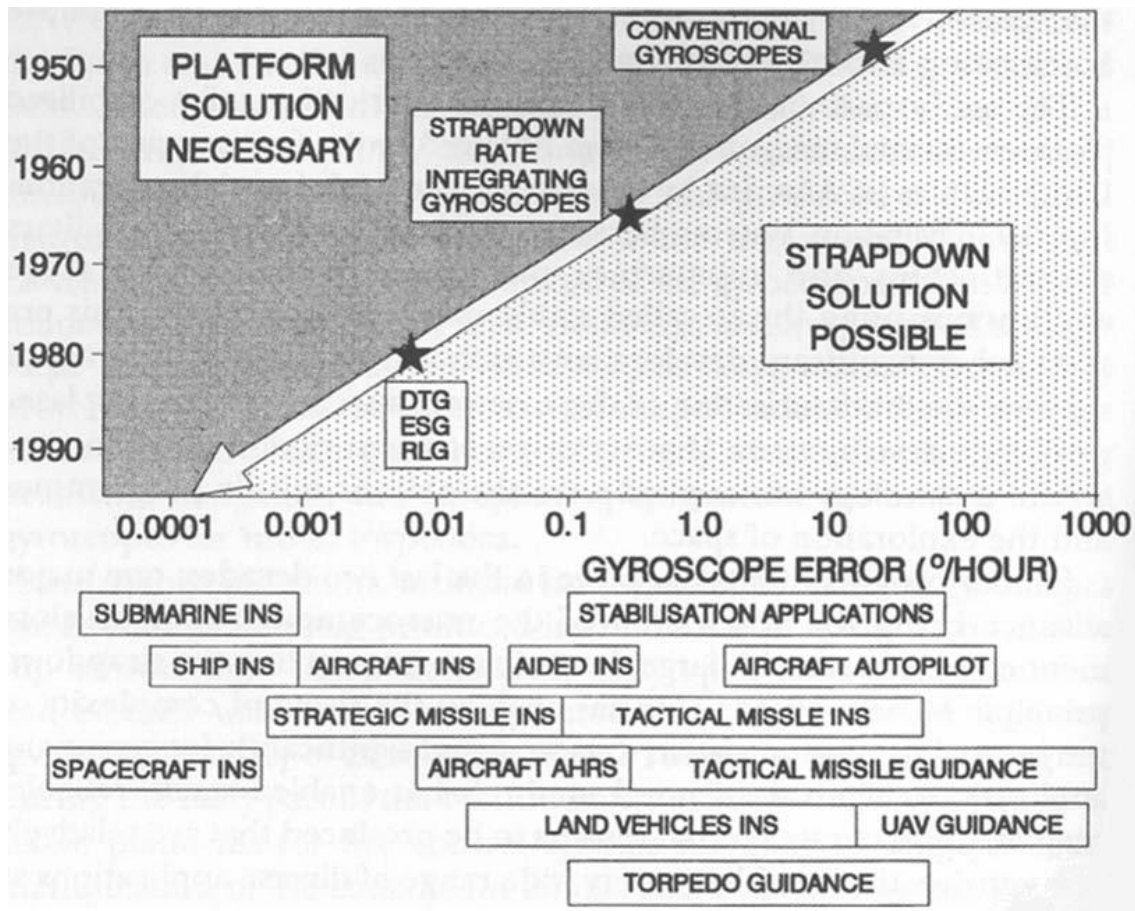


Figure 2.2 INS Implementations [33]

2.3.3 INS Errors. The two main INS error sources for a low-quality INS are initial alignment error and inertial sensor errors. Once an INS has an error in its DCM relating the vehicle's orientation with respect to the navigation frame (whether from incorrect rotation measurements or from an alignment error), gravity is not correctly removed from the specific force measurements. This corrupts the measurements of what the INS "believes" is purely vehicle acceleration, and the system's position and velocity error grows. Because of the integration process used to keep track of position and velocity, even a small error can grow quickly. An unaided INS has no corrective mechanism, so its position solution can quickly reach a useless state. To give some perspective on the effect these errors can have on an INS

position solution, an INS in a commercial aircraft, which has to provide a meaningful solution over a period of several hours, has an error on the order of 1 nautical mile of error growth per hour [33]. This error growth rate is considered better than average INS performance (but is not as accurate as INS used in submarines or spacecraft).

Two common inertial sensor errors are noise and bias. Obviously, a high noise on any instrument makes its individual measurements unreliable. An instrument's bias directly causes error as well. An accelerometer's bias is the amount of acceleration that is measured in the absence of any true acceleration. This bias feeds directly into the measurement of specific forces, and is a large contributor to error growth. A gyroscope's bias is the rotation rate it measures in the absence of any true rotation. This is also a large contributor to error growth, because it introduces errors directly into the DCM. However, high noise and bias levels are not the true problem, because noise can be removed by averaging, and if known, the bias can be removed from each measurement. One big potential problem, particularly with gyroscopes, is a lack of turn-on stability. Turn-on stability refers to how stable a gyroscope's bias is over time, as the instrument is turned on and off. If this bias is unknown from one turn-on to the next it will be difficult to remove from the sensor's measurements. Another big problem can be caused by a bias that changes unpredictably during operation, making it very hard to estimate and remove from incoming measurements.

2.3.3.1 MEMS Sensor Errors. MEMS inertial sensors, which are examined in this research, generally can not approach the quality of non-MEMS sensors. MEMS accelerometers provide measurements that are good enough to make MEMS gyroscopes the limiting factor in INS performance. The majority of MEMS accelerometers work by using a proof mass that is displaced by acceleration along a certain axis, with the displacement of the proof mass being detected by measuring the change in capacitance between the proof mass and a non-moving electrode [15]. Figure 2.3 shows a cross-section of a MEMS accelerometer [2], with the proof mass suspended between two sensing electrodes.

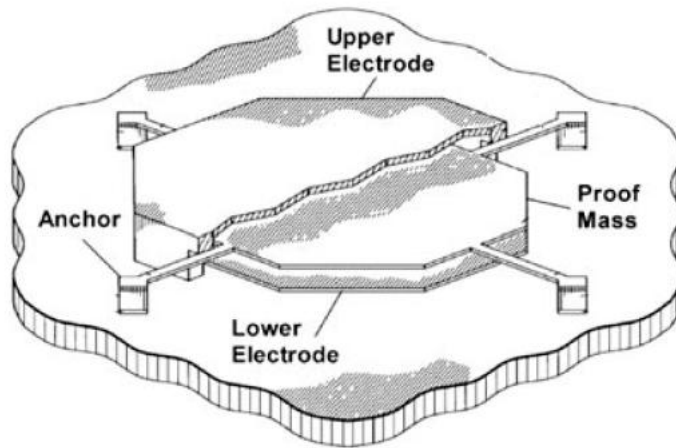


Figure 2.3 A MEMS Accelerometer [2]

MEMS gyroscopes are the source of the majority of the error in MEMS navigation applications. MEMS gyroscopes use the same principal as traditional mechanical gyroscopes, the Coriolis effect [33], to measure angular rotation. They do this by using a vibrating construct, similar to a tuning fork, to sense angular rotation and, as MEMS accelerometers do, using a change in capacitance to detect the magnitude of the rotation [5]. Figure 2.4 shows a photograph of a MEMS gyroscope [21], with the vibrating mass in the center of four electrodes.

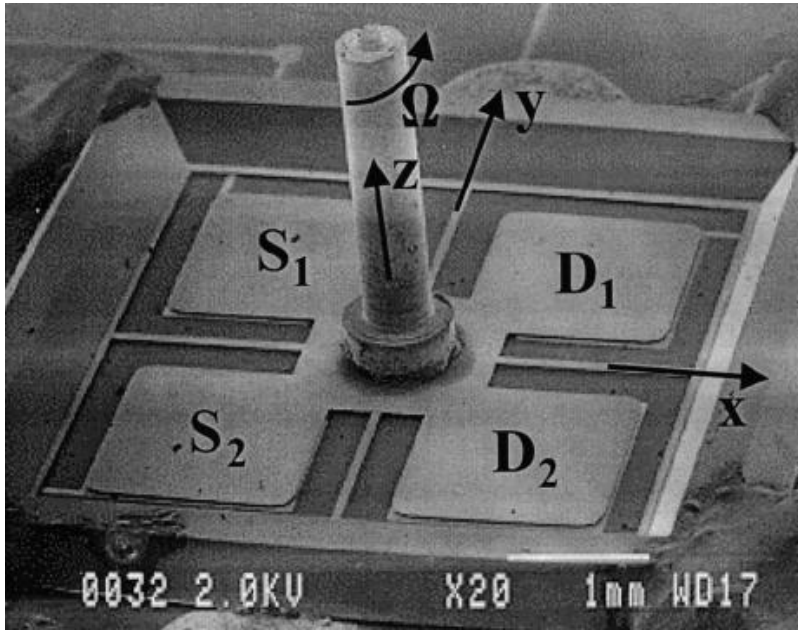


Figure 2.4 A Vibratory MEMS Gyroscope [21]

Both MEMS gyroscopes and accelerometers typically have much higher noise and bias levels than non-MEMS instruments, not to mention greater problems with bias instability. Two large sources of error are electrical noise and mechanical noise [15]. Electrical noise is the noise of the detection circuitry, and mechanical noise is caused by the Brownian motion of the air [15]. Temperature changes can also cause accuracy problems. Because MEMS sensors rely on the mechanical characteristics of the sensing materials, any changes in temperature that cause the materials to react differently can cause large errors. Also, it is readily apparent that because of MEMS small size, forces that do not cause problems for larger instruments can cause major errors. Electrical or magnetic fields, as well as forces on the atomic level, are all error sources that make MEMS sensors much less accurate than their larger counterparts. Much research is ongoing to improve MEMS gyroscope performance. However, the motivation for this research is to examine the accuracy potential of current commercially available MEMS inertial sensors.

2.4 Global Positioning System

The Global Positioning System, as mentioned previously, consists of a number of satellites that orbit the earth, monitored and operated by a network of ground stations. This network of satellites transmits signals meant for both military and civilian use. These signals are captured by a receiver, and used to calculate the receiver's position. The code generally used by civilians, called C/A code (for Coarse/Acquisition), provides the measurements that are used in this research.

2.4.1 GPS Principle of Operation [23]. The basic principle of operation (see Figure 2.5) is that each GPS satellite sends out a navigation signal, along with a set of orbital parameters called ephemeris data. A GPS receiver captures this data, and can use the ephemeris to calculate the position of the satellite at any point in time during a four-hour window. (The ephemeris data each satellite transmits is updated regularly by the ground stations that monitor the satellites.) The navigation signal, time-stamped with the satellite's time of transmission, is used to calculate the receiver's range to the satellite. This is done by multiplying the time difference between the signal reception and the signal transmission by the speed of light. This measurement of range is called pseudorange because it is not a true range: the receiver's clock is not perfectly synchronized with the satellite's clock, and this causes an error in the receiver's time of signal reception. With a minimum of four GPS pseudorange measurements, a receiver can calculate its position. Four satellites are needed for a position solution, because four variables are solved: the position of the receiver in three dimensions, and the clock error of the receiver (which is needed to remove the large error it induces in the pseudoranges).

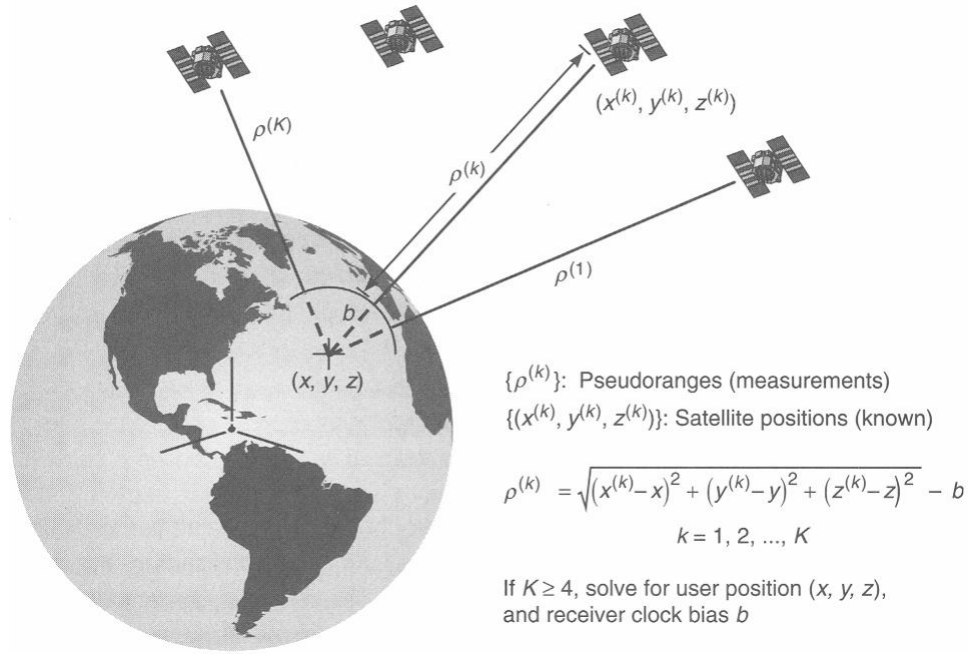


Figure 2.5 Satellite Navigation [23]

2.4.2 Errors in the Pseudorange Measurement. The pseudorange can be described as the true range between the satellite and receiver plus the effects of a number of error sources [6]. This is expressed as: [6] [31]

$$\rho = r + \Delta\rho + c(\delta t_u - \delta t_{sv}) + T + I + m_\rho + \epsilon_\rho \quad (2.1)$$

where

ρ = GPS pseudorange measurement (meters)

r = true range from the user to satellite (meters)

$\Delta\rho$ = satellite ephemeris error (meters)

c = speed of light (meters/second)

δt_u = receiver (user) clock error (seconds)

δt_{sv} = transmitter (space vehicle) clock error (seconds)

T = errors due to tropospheric delay (meters)

I = errors due to ionospheric delay (meters)

m_ρ = errors due to pseudorange multipath (meters)

ϵ_ρ = receiver errors - noise, interchannel bias, quantization, etc. (meters)

The satellite's transmitted ephemeris causes an error, because the orbital parameters contained in the ephemeris provide a close approximation, not a perfect solution, of the satellite's true position. The receiver clock error (δt_u) is calculated and its effects are removed from the final solution. The satellite clock, though extremely accurate, also induces a small error (δt_{sv}). There are also errors caused by two different parts of the atmosphere, the troposphere and the ionosphere. The troposphere consists of all non-charged atmospheric particles, and the error it causes (T) can be estimated using a model that takes the relative humidity as an input. The ionosphere, which consists of ionized particles, creates problems that are difficult to correct by using modeling, and much of its error (I) depends upon the time of day and the amount of ultraviolet light coming from the sun. Ionospheric error can be totally removed using dual frequency measurements, because the two GPS frequencies are affected by the ionosphere in different ways, allowing the ionospheric error to be isolated. Multipath (m_ρ) is simply the error induced by the satellite signal reflecting off obstacles near the receiver (buildings, trees, vehicles, etc.) The receiver error due to circuit noise,

quantization, etc., is all lumped together in the last term (ϵ_ρ). A basic estimate of the error caused by each of these sources is shown in Table 2.1 (not including user clock error, since its effects are removed).

Table 2.1 Approximate Magnitudes of Pseudorange Errors [18] [25] [28] [32]

Error Source	Pseudorange Error Magnitude - C/A code (m)
Transmitter clock error	2.1
Satellite ephemeris error	1.2
Ionospheric delay error	1 – 30
Tropospheric delay error	2 – 8
Receiver errors	0.5
Multipath error	1.4

The largest error, ionospheric error, is very volatile and hard to predict, although its effects tend to be reduced at night. To address the other large error, tropospheric delay, the basic modeling function mentioned previously can be used to correct for the majority of this error. With only this correction done, a basic GPS receiver, using pseudorange measurements, can reasonably achieve an accuracy of 5 to 7.5 meters RMS [23].

2.5 Kalman Filters

A Kalman filter is a technique based on stochastic system modeling that can be applied to a controls or data processing problem when deterministic models and techniques are not sufficient. There are several reasons why a deterministic model would be inadequate [20]. Firstly, no mathematical model of a system is perfect; a stochastic system model provides some allowances for system variations and unmodeled effects. Secondly, dynamic systems are driven not only by control inputs, but by internal and external disturbances that cannot be modeled or controlled deterministically. Thirdly, system sensors do not provide perfectly accurate data. Data

is corrupted by noise, and sensors provide incomplete information and have inherent errors such as bias or non-linearity. Stochastic modeling is much more effective than deterministic techniques when dealing with these system realities.

A Kalman filter is best described as an “optimal recursive data processing algorithm” [20]. It is optimal in many ways, one of which is that it uses all available data, regardless of its quality. It is recursive in that it takes into account the effects of all previous data, *without* requiring all old data to be stored and re-processed every time new data is available. It is also a data processing algorithm, usually in the form of a computer program run by a central processor, not a “black-box” electrical filter.

One main assumption provides the basis for the Kalman filter [6]. This assumption is that an adequate model of the real-world system exists (in the form of a linear dynamics model), from which linear measurements are taken. With a poor model of the actual real-world system, a Kalman filter will not perform well. This system model is assumed to be driven by white Gaussian noise of known statistics, and the measurements are also corrupted by white Gaussian noise of known statistics. The complete set of models that the filter needs consists of a system dynamics model, the various measurement models for system sensors, and the stochastic models for model uncertainties, measurement noises and errors, and system noises [6].

Because all Kalman filters are implemented using a digital computer, a sampled-data version of the filter is used. This sampled-data Kalman filter uses a propagate-update cycle. The propagation cycle takes the filter’s estimate of the system state from a previous sample time and provides a new estimate of the system state at the current time, based upon the dynamics model. The update cycle occurs when new measurements are available and updates the system state based upon the new measurements and the system measurement model. A complete derivation of the Kalman filter may be found in [20].

2.5.1 State Model Equations. The development in this and the following sub-section is largely based on [6], which is itself based on [16] and [20]. Upper case bold letters indicate matrices, lower case bold letters indicate vectors, and normal or italics represent scalar variables. Random vectors are denoted by boldface sans serif type.

The system dynamics of the real-world system are modeled linearly, with a state differential equation of the form

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{G}(t)\mathbf{w}(t) \quad (2.2)$$

where

$\mathbf{x}(t)$ = the n-dimensional system state vector

$\mathbf{F}(t)$ = the n-by-n system dynamics matrix

$\mathbf{B}(t)$ = the n-by-r control input matrix

$\mathbf{u}(t)$ = the r-dimensional control input

$\mathbf{G}(t)$ = the n-by-s noise input matrix

$\mathbf{w}(t)$ = the s-dimensional dynamics driving noise vector

The noise vector $\mathbf{w}(t)$ is white and Gaussian with a strength of $\mathbf{Q}(t)$ and statistics

$$E\{\mathbf{w}(t)\} = \mathbf{0} \quad (2.3)$$

$$E\{\mathbf{w}(t)\mathbf{w}^T(t')\} = \mathbf{Q}(t)\delta(t - t') \quad (2.4)$$

(In this research, no control inputs need to be modeled, so the \mathbf{B} and \mathbf{u} terms will be dropped from any subsequent equations.) At discrete times, the solution to Equation

2.2 can be written as:

$$\mathbf{x}(t_{i+1}) = \Phi(t_{i+1}, t_i)\mathbf{x}(t_i) + \left[\int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, \tau) \mathbf{G}(\tau) d\boldsymbol{\beta}(\tau) \right] \quad (2.5)$$

where $\boldsymbol{\beta}$ is a vector-valued Brownian motion process of diffusion $\mathbf{Q}(t)$ [20], and where $\Phi(t_{i+1}, t_i)$ is the state transition matrix from time t_i to time t_{i+1} . This state transition matrix (assuming a time-invariant \mathbf{F} matrix) is given by

$$\Phi(t_{i+1}, t_i) = \Phi(\Delta t) = e^{\mathbf{F}\Delta t} \text{ where } \Delta t \equiv t_{i+1} - t_i \quad (2.6)$$

The equivalent discrete-time model for Equation 2.2 is expressed by the stochastic difference equation

$$\mathbf{x}(t_{i+1}) = \Phi(t_{i+1}, t_i)\mathbf{x}(t_i) + \mathbf{w}_d(t_i) \quad (2.7)$$

where

$$\mathbf{w}_d(t_i) = \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, \tau) \mathbf{G}(\tau) d\boldsymbol{\beta}(\tau) \quad (2.8)$$

The discrete-time white Gaussian dynamics driving noise ($\mathbf{w}_d(t_i)$) has the statistics:

$$E\{\mathbf{w}_d(t_i)\} = \mathbf{0} \quad (2.9)$$

$$E\{\mathbf{w}_d(t_i)\mathbf{w}_d^T(t_i)\} = \mathbf{Q}_d(t_i) = \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, \tau) \mathbf{G}(\tau) \mathbf{Q}(\tau) \mathbf{G}^T(\tau) \Phi^T(t_{i+1}, \tau) d\tau \quad (2.10)$$

$$E\{\mathbf{w}_d(t_i)\mathbf{w}_d^T(t_j)\} = \mathbf{0}, t_i \neq t_j \quad (2.11)$$

2.5.2 Measurement Model Equations. The real-world systems and problems to which Kalman filters are usually applied can be defined by continuous-time dynamics processes. Sensors then produce sampled-data measurements which are modeled as a linear, discrete-time equation of the form

$$\mathbf{z}(t_i) = \mathbf{H}(t_i)\mathbf{x}(t_i) + \mathbf{v}(t_i) \quad (2.12)$$

where $\mathbf{z}(t_i)$ is the modeled measurement, $\mathbf{H}(t_i)$ is the measurement model matrix, and $\mathbf{v}(t_i)$ is the measurement noise, with statistics

$$E\{\mathbf{v}(t_i)\} = \mathbf{0} \quad (2.13)$$

$$E\{\mathbf{v}(t_i)\mathbf{v}^T(t_j)\} = \begin{cases} \mathbf{R}(t_i) & \text{for } t_i = t_j \\ \mathbf{0} & \text{for } t_i \neq t_j \end{cases} \quad (2.14)$$

The dynamics driving noise $\mathbf{w}_d(t_i)$ and the measurement corruption noise $\mathbf{v}(t_i)$ are assumed to be independent, so

$$E\{\mathbf{w}_d(t_i)\mathbf{v}^T(t_j)\} = \mathbf{0} \text{ for all } t_i \text{ and } t_j \quad (2.15)$$

2.5.3 System Propagate and Update Equations. As previously mentioned, a discrete-time Kalman filter uses a propagate-update cycle to provide its system estimates. The filter must be provided initial conditions for the system states, in the form of the vector $\hat{\mathbf{x}}(t_0)$, and the state covariances, with the matrix $\mathbf{P}(t_0)$. Once these are provided, the filter can start propagating forward in time until the first update cycle. The propagation equations are: [20]

$$\hat{\mathbf{x}}(t_i^-) = \Phi(t_i, t_{i-1})\hat{\mathbf{x}}(t_{i-1}^+) \quad (2.16)$$

$$\mathbf{P}(t_i^-) = \Phi(t_i, t_{i-1})\mathbf{P}(t_{i-1}^+)\Phi^T(t_i, t_{i-1}) + \mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1}) \quad (2.17)$$

where the superscript “+” represents the filter state and covariance estimates after an update cycle, and the superscript “-” denotes the filter states prior to an update cycle.

When sensor measurements become available, they update the state estimates using the following equations: [20]

$$\mathbf{K}(t_i) = \mathbf{P}(t_i^-) \mathbf{H}^T(t_i) [\mathbf{H}(t_i) \mathbf{P}(t_i^-) \mathbf{H}^T(t_i) + \mathbf{R}(t_i)]^{-1} \quad (2.18)$$

$$\mathbf{r}(t_i) = \mathbf{z}_i - \mathbf{H}(t_i) \hat{\mathbf{x}}(t_i^-) \quad (2.19)$$

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i) \mathbf{r}(t_i) \quad (2.20)$$

$$\mathbf{P}(t_i^+) = \mathbf{P}(t_i^-) - \mathbf{K}(t_i) \mathbf{H}(t_i) \mathbf{P}(t_i^-) \quad (2.21)$$

The matrix $\mathbf{K}(t_i)$ is the Kalman filter “gain”, and it determines how much the filter relies on its own internal dynamics model versus relying on new measurements. $\mathbf{K}(t_i)$ is directly affected by the values set in the measurement noise matrix \mathbf{R} . The vector $\mathbf{r}(t_i)$ is called the residual vector, and it represents the difference between the actual measurement values and the filter-predicted measurements.

“Tuning” a filter involves adjusting the dynamics process noise strength \mathbf{Q} and the measurement noise covariance \mathbf{R} for the best filter performance. A filter’s performance can be measured by how well it models and predicts the performance of the real-world system, and by how well its covariance matrix \mathbf{P} , which gives the filter’s estimate of its own accuracy, matches the filter’s actual accuracy. A well-designed, well-tuned filter will have residuals that are zero-mean, white, and Gaussian in nature (and of covariance $[\mathbf{H}(t_i) \mathbf{P}(t_i^-) \mathbf{H}^T(t_i) + \mathbf{R}(t_i)]$), indicating that the filter is making the best possible use of the measurements.

2.6 Test Equipment

2.6.1 XSens MT-9B. Because the ultimate goal of this research is an actual real-time system, extensive research was accomplished on MEMS IMUs. The MEMS IMU chosen for actual data collection was the XSens MT-9B [39] (Figure 2.6). The MT-9 was initially chosen for its price (at the time of purchase, its cost was approximately \$1,500 US) and for ease of acquisition. Although there were other

MEMS IMUs available, most were significantly more expensive or very difficult to purchase for secrecy or supply reasons. The MT-9 was compared to another IMU, the MicroStrain 3DM-G [22], and was chosen because it had much less quantization and drift with its gyroscope measurements.

XSens is a small Netherlands-based company that designed the MT-9 specifically for body-tracking applications, although its navigation potential was recognized. The MT-9 contains nine MEMS sensors: three each of gyroscopes, accelerometers, and magnetometers. It can sense up to 900 degrees/second of rotation and 100 meters/second² of acceleration. Its output options include raw and calibrated data, all time-stamped with an internal counter. The raw data consists of the sampled measurements converted to digital form, and the calibrated data is compensated for instrument misalignment and bias (both measured in the factory). It is capable of output rates up to 256 Hz, and its power requirements are low (220 mW). It is very small (approximately 3.6 cubic inches) and weighs only 35 grams. The output interface uses the common RS-232 serial standard.



Figure 2.6 XSens MT-9B IMU

2.6.2 NovAtel Black Diamond System. To provide a reference system with which to compare the MEMS system, the NovAtel Black Diamond System (BDS) was used. The BDS is an Kalman filter-integrated INS/GPS that provides 100

Hz output data. The IMU used in the BDS is a Honeywell HG-1700. The HG-1700 uses ring-laser gyroscopes, which are much more accurate than the MT-9's MEMS gyroscopes. After it was purchased, the IMU's gyroscopes were tested and found to have approximately 0.1 degrees/hour drift (see Figure 2.2), with an overall performance in the tactical-grade level (a tactical-grade INS would be accurate over a period of minutes). This level of performance makes it ideal to provide the truth data for the MEMS-based system. The BDS is currently out of production, but at the time of purchase the cost was well over \$30,000 US.



Figure 2.7 NovAtel Black Diamond System (INS/GPS)

2.7 Summary

This chapter provided a basic overview of INS and GPS principles, operation, and errors. The Kalman filter was described along with the appropriate state, measurement model, and propagate and update equations. Finally, an overview was given of the test equipment used to provide data.

III. System Development

3.1 Overview

This chapter contains a detailed description of the algorithm that implements this tightly-coupled INS/GPS integration. This includes descriptions of the system states, dynamics model, and measurement model, and also provides necessary details about the software structure.

3.2 Algorithm

3.2.1 Filter states. The Kalman filter used in this research is an “error-state” implementation. This means that the filter does not explicitly estimate the variables of interest (in this case, position, velocity, and attitude), but rather it estimates the *errors* in the INS indicated position, velocity, and attitude. An error-state model is useful because even in a high-dynamic environment, where position, velocity, and attitude are rapidly changing, the *errors* in these states change slowly when compared to the high frequency dynamics. The slowly-changing nature of the estimated variables enhances filter stability [20], providing better overall performance. The seventeen states implemented in the Kalman filter are listed following:

$x_1 = \delta lat$ = error in INS indicated latitude (degrees)

$x_2 = \delta lon$ = error in INS indicated longitude (degrees)

$x_3 = \delta alt$ = error in INS indicated altitude (meters)

$x_4 = \delta V_N$ = error in INS indicated North velocity (meters/second)

$x_5 = \delta V_E$ = error in INS indicated East velocity (meters/second)

$x_6 = \delta V_D$ = error in INS indicated Down velocity (meters/second)

$x_7 = \delta \alpha$ = error in INS indicated tilt about the x-axis (roll) (radians)

$x_8 = \delta\beta$ = error in INS indicated tilt about the y-axis (pitch) (radians)

$x_9 = \delta\gamma$ = error in INS indicated tilt about the z-axis (yaw) (radians)

$x_{10} = \delta t$ = GPS receiver clock bias (meters)

$x_{11} = \delta\dot{t}$ = GPS receiver clock drift (meters/second)

$x_{12} = bias_{xAccel}$ = x-axis accelerometer bias (meters/second²)

$x_{13} = bias_{yAccel}$ = y-axis accelerometer bias (meters/second²)

$x_{14} = bias_{zAccel}$ = z-axis accelerometer bias (meters/second²)

$x_{15} = bias_{xGyro}$ = x-axis gyroscope bias (radians/second)

$x_{16} = bias_{yGyro}$ = y-axis gyroscope bias (radians/second)

$x_{17} = bias_{zGyro}$ = z-axis gyroscope bias (radians/second)

In addition to the nine fundamental INS error states, the filter explicitly models eight instrument errors: two GPS receiver clock states, three accelerometer biases, and three gyroscope biases (even though the gyroscope's output is an angular rate, there is a bias in this rate output). The clock states have units of meters and meters/second because it is numerically simpler to model these states in terms of distance and speed rather than time. The decision to model eight instrument errors was made because it was hoped that this would provide the system with better performance at a minimal cost in processing time. Chapter 4 presents a deeper discussion of the results of this decision.

3.2.2 System Dynamics Model. The system dynamics model provides the Kalman filter with its “picture” of how the system is expected to behave over time. This model consists of the **F** matrix and the **Q** matrix. The **F** matrix relates the time derivatives of each system state to the system state vector, in the manner shown by Equation 2.2. The model for the first nine states (time derivatives of errors in position, velocity, and attitude) comes from the Pinson error model, a well-documented and tested dynamics model which is discussed in detail in [30] and [33].

The Pinson model for the velocity error states is augmented by using a body-to-navigation frame DCM to relate the velocity states to the accelerometer bias states. Similarly, the attitude error states are related to the gyroscope bias states using a body-to-navigation frame DCM. The equations for the Pinson error model are shown following in Equation 3.1. The full \mathbf{F} matrix is shown in Equation 3.2.

$$\mathbf{F}_{\text{Pinson}} = \begin{bmatrix} 0 & 0 & \frac{-V_N}{R^2} & \frac{1}{R} & 0 & 0 & 0 & 0 & 0 \\ \frac{V_E \tan L}{R \cos L} & 0 & \frac{-V_E}{R^2 \cos L} & 0 & \frac{1}{R \cos L} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ F_{41} & 0 & F_{43} & \frac{V_D}{R} & F_{45} & \frac{V_N}{R} & 0 & -f_D & f_E \\ F_{51} & 0 & F_{53} & F_{54} & F_{55} & F_{56} & f_D & 0 & -f_N \\ 2\Omega V_E \sin L & 0 & F_{63} & \frac{-2V_N}{R} & F_{65} & 0 & -f_E & f_N & 0 \\ -\Omega \sin L & 0 & \frac{-V_E^2}{R^2} & 0 & \frac{1}{R} & 0 & 0 & F_{78} & \frac{V_N}{R} \\ 0 & 0 & \frac{V_N}{R^2} & -\frac{1}{R} & 0 & 0 & F_{87} & 0 & F_{89} \\ F_{91} & 0 & \frac{V_E \tan L}{R^2} & 0 & \frac{-\tan L}{R} & 0 & -\frac{V_N}{R} & F_{98} & 0 \end{bmatrix} \quad (3.1)$$

where

$$F_{41} = -V_E \left(2\Omega \cos L + \frac{V_E}{R \cos^2 L} \right)$$

$$F_{43} = \frac{1}{R^2} (V_E^2 \tan L - V_N V_D)$$

$$F_{45} = -2 \left(\Omega \sin L + \frac{V_E}{R} \tan L \right)$$

$$F_{51} = 2\Omega (V_N \cos L - V_D \sin L) + \frac{V_N V_E}{R \cos^2 L}$$

$$F_{53} = -\frac{V_E}{R^2} (V_N \tan L + V_D)$$

$$F_{54} = 2\Omega \sin L + \frac{V_E}{R} \tan L$$

$$F_{55} = \frac{1}{R} (V_N \tan L + V_D)$$

$$F_{56} = 2\Omega \cos L + \frac{V_E}{R}$$

$$F_{63} = \frac{1}{R^2}(V_N^2 + V_E^2)$$

$$F_{65} = -2 \left(\Omega \cos L - \frac{V_E}{R} \right)$$

$$F_{78} = -\Omega \sin L - \frac{V_E}{R} \tan L$$

$$F_{87} = \Omega \sin L + \frac{V_E}{R} \tan L$$

$$F_{89} = \Omega \cos L + \frac{V_E}{R}$$

$$F_{91} = -\Omega \cos L - \frac{V_E}{R \cos^2 L}$$

$$F_{98} = -\Omega \cos L - \frac{V_E}{R}$$

and where R is the radius of the earth, V_N , V_E , and V_D are the INS indicated north, east, and down velocities, f_N , f_E , and f_D are the IMU's north, east, and down specific force measurements, L is the INS indicated Latitude, and Ω is the earth's rate of rotation.

$$\mathbf{F}_{17 \times 17} = \begin{bmatrix}
& & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
& & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
& & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
& & & & & & & & & & 0 & 0 & & 0 & 0 & 0 & 0 & 0 \\
& & & & & F_{Pinson-9 \times 9} & & & & & 0 & 0 & DCM_{3 \times 3} & 0 & 0 & 0 & 0 & 0 \\
& & & & & & & & & & 0 & 0 & & 0 & 0 & 0 & 0 & 0 \\
& & & & & & & & & & 0 & 0 & 0 & 0 & 0 & & & \\
& & & & & & & & & & 0 & 0 & 0 & 0 & 0 & DCM_{3 \times 3} & & \\
& & & & & & & & & & 0 & 0 & 0 & 0 & 0 & & & \\
& & & & & & & & & & 0 & 0 & 0 & 0 & 0 & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
& & & & & & & & & & & & & & & & & \mathbf{0}_{6 \times 17}
\end{bmatrix} \quad (3.2)$$

States ten through seventeen use fairly simple dynamics models. The derivative of state ten, the GPS receiver clock bias state, is directly modeled by state eleven, the GPS receiver clock drift state (represented by a “1” in the tenth row, eleventh column). The derivatives of the last ten states are represented by using a random-walk model [20], which does not have a component in the \mathbf{F} matrix. Mathematically, a random-walk model is the output of an integrator driven by an input of white Gaussian noise [20]. This is represented in the Kalman filter by allowing the modeled state to be driven purely by the value set in the noise matrix \mathbf{Q} . The intent is that the filter will quickly assign a (hopefully correct) error value to the state, and then allow this value to vary slowly or quickly depending on the corresponding value in \mathbf{Q} [20]. Ideally, each instrument’s assigned \mathbf{Q} value will match the instrument’s actual noise characteristics.

3.2.3 Measurement Model. The measurement model, contained in the \mathbf{H} matrix, relates the measurements to the filter states. The model chosen for this implementation uses the GPS pseudorange minus what can be called the “INS pseudorange” as the measurement \mathbf{z} . The INS pseudorange is simply the range to the satellite as calculated from the INS-indicated position. (Each satellite’s position is known from the ephemeris data it transmits.) By forming the measurements this way, the states are linear with respect to the measurements, and a standard linear Kalman filter measurement model can be used. Equations 3.3 and 3.4, shown following, describe the various components of the INS and GPS pseudorange:

$$\rho_{gps} = \rho_{true} + \delta t + v \quad (3.3)$$

$$\rho_{ins} = \rho_{true} + \delta\rho_{ins} \quad (3.4)$$

The GPS pseudorange is modeled as the true range (ρ_{true}) plus the effects of GPS clock error (δt) plus measurement noise v . The INS pseudorange is modeled as the true range (ρ_{true}) plus INS range error ($\delta\rho_{ins}$), which consists of the effects of INS position drift on the range between the INS and the satellite. The measurement vector \mathbf{z} is then calculated as shown in the following equation:

$$z = \rho_{gps} - \rho_{ins} = \delta t - \delta\rho_{ins} + v \quad (3.5)$$

Equation 3.5 is of the form $\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v}$ (see Section 2.5.2), where

$$\mathbf{H}\mathbf{x} = \delta t - \delta\rho_{ins} \quad (3.6)$$

The right-hand side of Equation 3.6 is the quantity that the \mathbf{H} matrix must relate to the filter states. The δt portion of the equation is represented by state 10 in the \mathbf{x} vector, so this is easily represented by putting a “1” in the 10th column of the \mathbf{H} matrix. It is more complicated to relate $\delta\rho_{ins}$ to the states. See Figure 3.1 for

a diagram which will be referenced to explain how this is done. (Note: Figure 3.1 is greatly exaggerated for purposes of illustration. All quantities on the illustration are scalars unless they have a \rightarrow over them.)

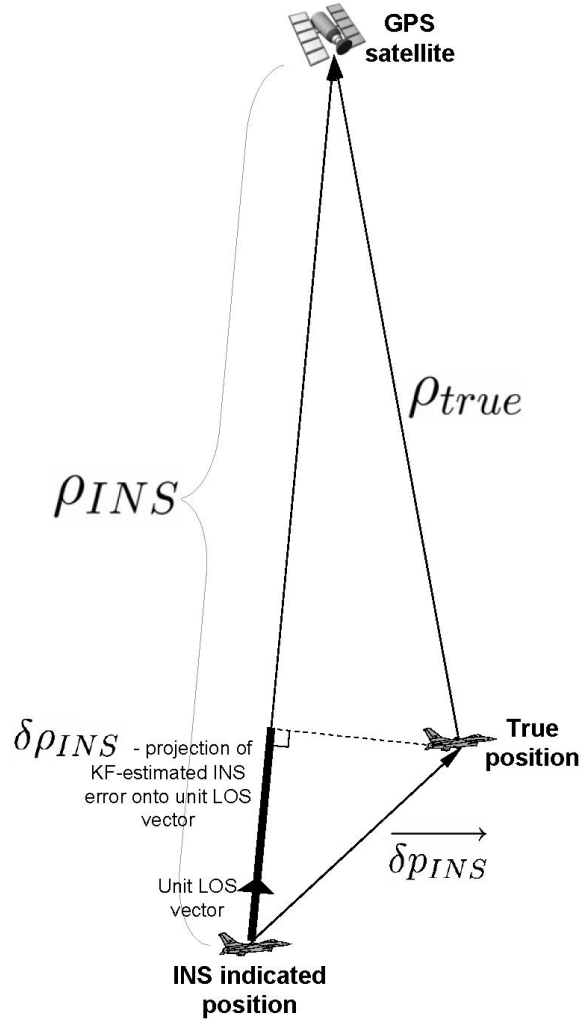


Figure 3.1 Filter Measurement Model

The pseudorange provided by the GPS receiver is assumed to be the true pseudorange (ρ_{true}). The line from the INS-indicated position to the GPS satellite is the INS pseudorange (ρ_{ins}). The darkened portion of this line, the INS range error

$(\delta\rho_{ins})$, is the desired measurement (it can be positive or negative, depending on where the INS-indicated position is in relation to the true position). This quantity can be obtained by taking the projection of the INS position error vector ($\overrightarrow{\delta p_{ins}}$, obtained from the first three states in the state vector) onto the unit line-of-sight vector from the INS to the satellite. Note that this model assumes that the lighter portion of ρ_{ins} is the same length as the true range (ρ_{true}), which is a good assumption as long as the INS position error ($\overrightarrow{\delta p_{ins}}$) does not grow too large (it would take an error of tens of kilometers or more for this assumption to cause problems). The equation to relate the states to the measurements and perform this projection (which is a dot product operation, with some reference frame conversions added) is shown following.

$$\delta\rho_{ins} = e_{N(NED)}(R_n + alt_{ins})\delta lat + e_{E(NED)}(R_e + alt_{ins})\cos(lat_{ins})\delta lon - e_{D(NED)}\delta alt \quad (3.7)$$

The e terms are components of the unit line-of-sight vector from the INS to the satellite in the NED frame. The coefficients of the δlat , δlon , and δalt terms are needed to convert them to the NED frame (R_n and R_e are the osculating earth radii in the north and east directions, respectively, calculated using the current INS latitude). The last term of Equation 3.7 is negative because the NED frame has the opposite sign of the LLH frame in the vertical direction. This equation forms one row of the \mathbf{H} matrix, with one row for each pseudorange measurement from a GPS satellite. An example row of the \mathbf{H} matrix is shown following.

$$\begin{pmatrix} -e_{n(NED)}(R_n + alt_{ins}) & -e_{e(NED)}(R_e + alt_{ins})\cos(lat_{ins}) & e_{d(NED)} & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The signs are reversed from Equation 3.7 because in the \mathbf{Hx} equation (Equation 3.6), $\delta\rho_{ins}$ is subtracted from δt . The “1” in the 10th column represents the δt term of Equation 3.6.

The structure of the measurement model highlights an advantage of a tightly-coupled Kalman filter integration: even if only one satellite were available, GPS data could still be used to provide corrective measurements. This favorably contrasts with a loosely-coupled integration which uses the actual GPS position solution as a filter measurement, requiring a minimum of four satellites to be available to get any use from GPS.

3.3 Code Structure

Because the long-term goal of this thesis research is to pave the way for a real-time system, the filter software was written with code efficiency, speed, and software portability as the primary goals. Good code efficiency and speed will lower power and price requirements for the computing hardware, while portable code will allow the software to be run on a wide range of possible hardware configurations. To provide portability, the code was written using C++, with an AFIT contractor supplying code libraries for matrix math (C++ does not have native matrix libraries). Figure 3.2 is a flowchart of the code structure. It does not completely represent each calculation sequentially or temporally, but the overall concept is depicted accurately.

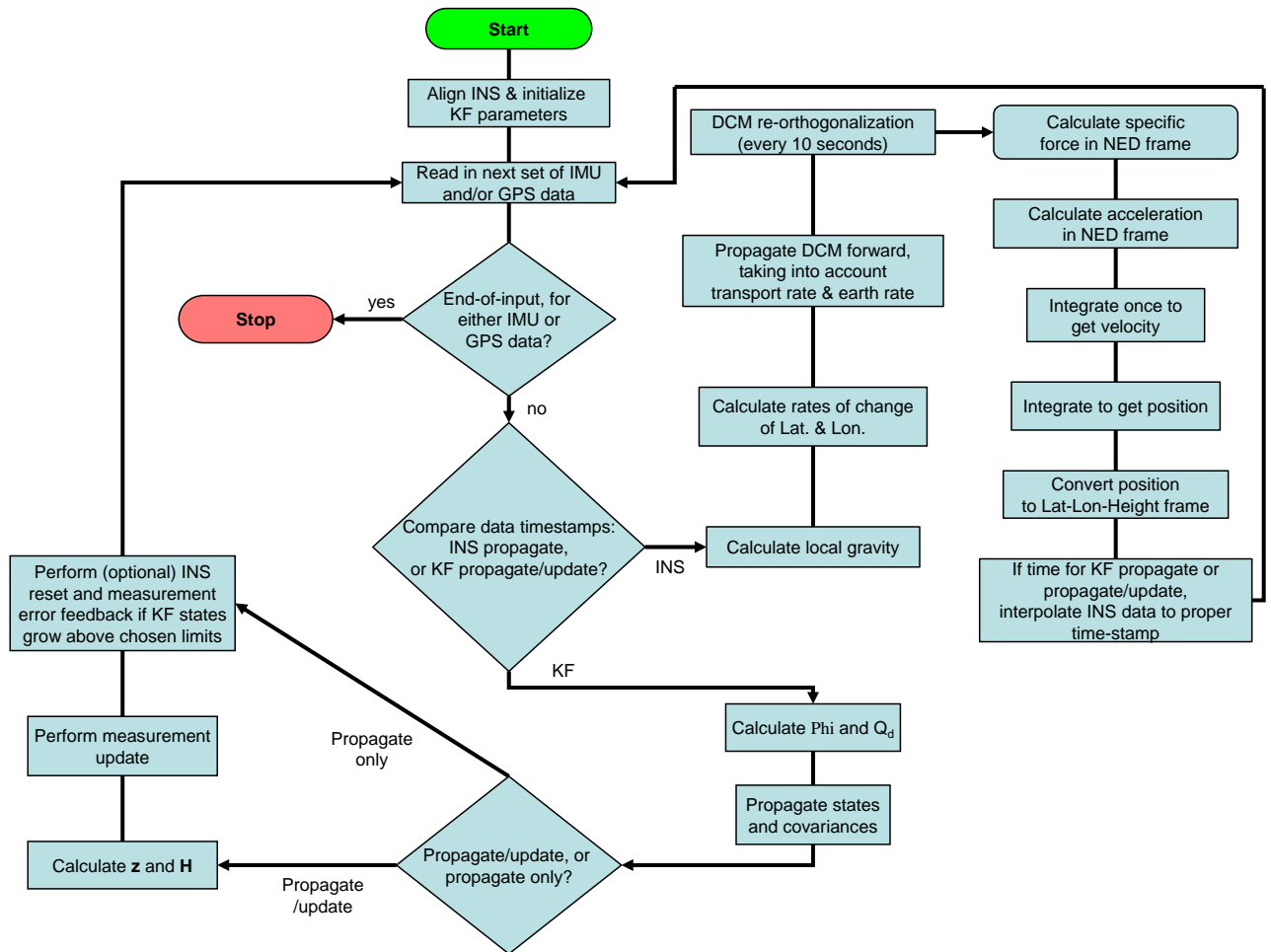


Figure 3.2 Code Flowchart

Because of the emphasis on a future real-time implementation, a number of compromises were made in the software. Firstly, although the INS provides its position solution at a 100 Hz rate, the Kalman filter only propagates and updates its error estimates at a 1 Hz rate (if measurement gaps exist in the 1 Hz GPS data, a propagate-only cycle is performed instead of a full propagate-update cycle). This means that the filter's error estimates are not kept current with the INS state; they are truly valid only at the actual update time, though they are used until the next filter cycle. This was deemed reasonable because the error states change relatively slowly, and should thus still be fairly accurate over a one second period,

even though they are not completely mathematically correct. This implementation trades minimal accuracy loss for a significant savings in processing time.

The dynamics matrix \mathbf{F} is affected by this disparity between the INS rate and the Kalman filter cycle rate. This matrix is re-calculated during each filter cycle, because it uses data taken directly from the INS state. However, the DCMs contained in the \mathbf{F} matrix (which relate velocity and attitude error states to instrument errors) can vary dramatically over one second (during movement); to use the most current INS-indicated DCM would not be a very accurate view of the entire one-second period. Therefore, *average* DCMs are calculated and used in the \mathbf{F} calculation. The average DCM is simply a mathematical mean; all of the DCMs between filter cycles are added up and then divided by the number of additions. Similarly, average values for the specific forces f_N , f_E , and f_D are also used when generating \mathbf{F} (again, the specific force averages are mathematical means). The rest of the INS values used in \mathbf{F} , such as the INS-indicated velocities and positions, use the current INS values. These last statements about INS values reveal an invalid assumption, namely that \mathbf{F} is time-invariant over the period between filter cycles. In reality, many INS values change over the interval between two \mathbf{F} calculations. However, this assumption is useful because it allows a linear Kalman filter to be used, and it does not cause serious problems because the golf cart used to collect data has fairly low dynamics. The other major compromises in the code are the use of first-order approximations to calculate the state-transition matrix Φ and the discrete-time noise matrix \mathbf{Q}_d . The full, mathematically correct equations for Φ and \mathbf{Q}_d (see Equations 2.6 and 2.10) use matrix exponential calculations, which are very processor-intensive. Thus, the following, much less processor-intensive, first-order equations were used [20]:

$$\Phi = \mathbf{I} + \mathbf{F}\Delta t \quad (3.8)$$

$$\mathbf{Q}_d = \mathbf{Q}\Delta t \quad (3.9)$$

where Δt is the time since the last filter propagation (this should be very close to one second in most cases). The use of these approximations was deemed reasonable because of the computing advantage, and also because the errors caused by this approximation were considered of a lower magnitude than the other errors introduced by using comparatively noisy and error-prone MEMS inertial instruments.

3.4 Summary

This chapter provided a system overview, starting with the Kalman filter's state description. The system dynamics model and measurement model were explained in detail. Lastly, the software code structure was described, with performance-related compromises specifically addressed.

IV. Testing and Analysis

4.1 Overview

This chapter describes the testing that was used to examine the design of the Kalman filter. Key filter features are described, and the final tuning parameters are listed and explained. Finally, the filter’s performance is examined over a wide range of scenarios including short-term accuracy, long-term accuracy, and GPS outages. Filter residuals are also shown and discussed.

To prevent confusion, terms are clarified as follows: when the term “MT-9” or “INS” is used, this refers to the overall system’s unaided INS, driven by the input from the MT-9 IMU. The term “filter” refers to the final system output: the MT-9 INS output with the Kalman filter’s error estimates applied to it.

Out of the five data sets created, three data sets are referenced in this chapter. “Figure Eights” is a short (less than 80 seconds) data set in which repeated figure eights were driven in a golf cart. “Short-term” is approximately 170 seconds in length. This is a data set in which the golf cart was driven with random direction changes and much stopping and starting. “Long-term” is slightly over 13.1 minutes. This data set encompasses all the other data sets, including several static time periods intended to provide starting points for the other data sets. Figures 4.1 through 4.3 show position tracks for these three data sets, and Table 4.1 provides a brief summary of this information. See Appendix A for the complete position, velocity, and attitude plots of these three data sets.

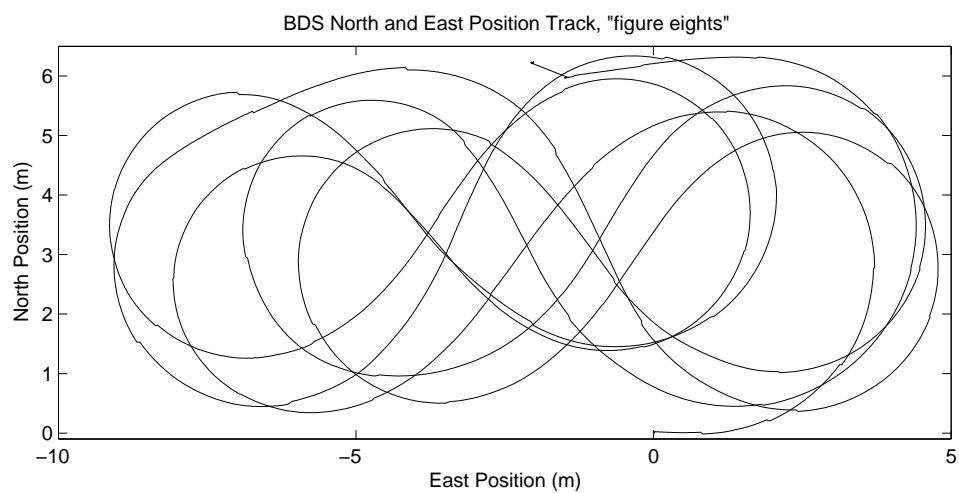


Figure 4.1 BDS North and East Position Track, “figure eights”

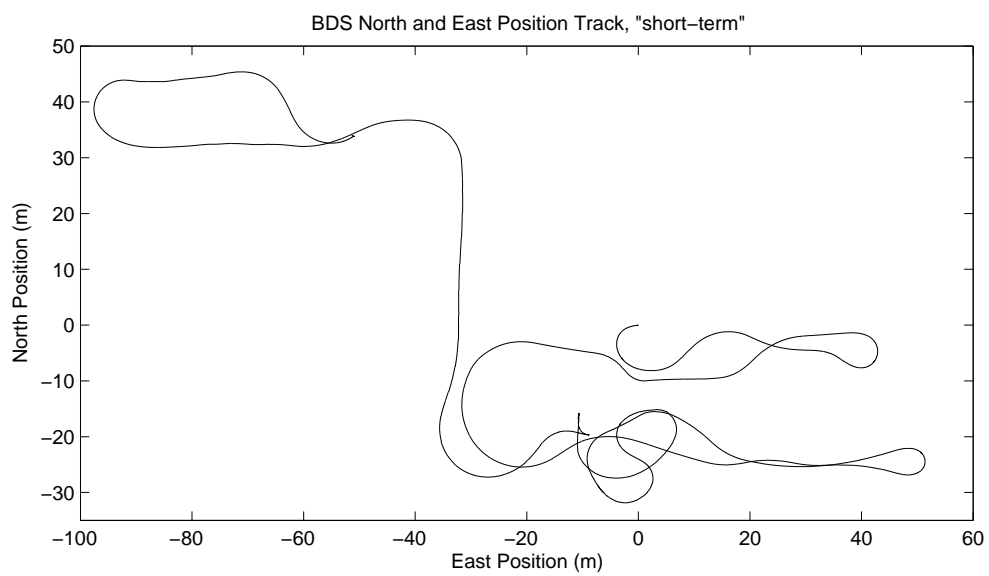


Figure 4.2 BDS North and East Position Track, “short-term”

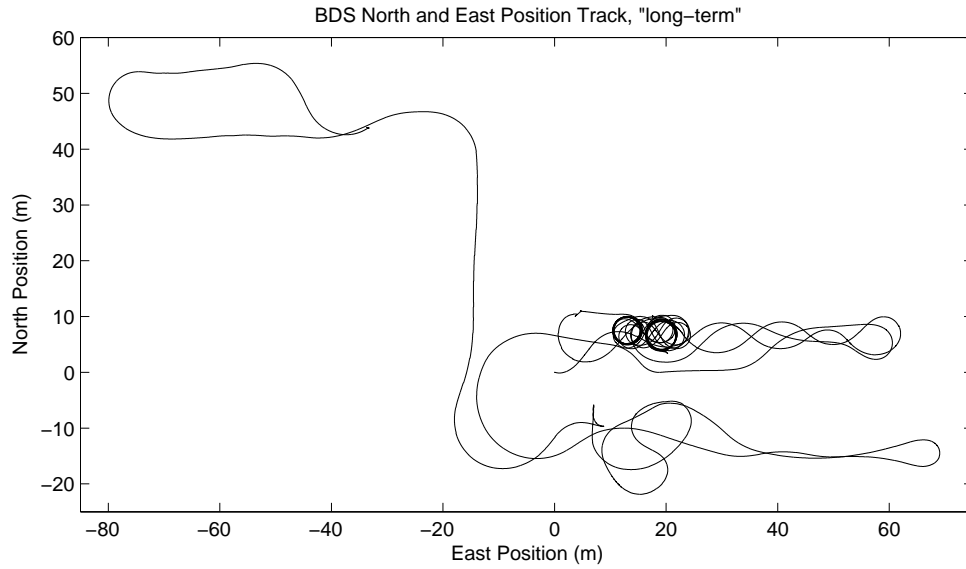


Figure 4.3 BDS North and East Position Track, “long-term”

Table 4.1 Data Set Summary

Name	Length	Description
Figure Eights	≈ 80 seconds	Repeated figure eights
Short-term	≈ 170 seconds	Random high-dynamic driving
Long-term	≈ 13.1 minutes	Entire data set

4.2 Data Methodology

In order to test the system filter design, experimental data needed to be gathered and processed. As discussed in Section 2.6.1, the MEMS IMU that was used in this research was the XSens MT-9B. The NovAtel Black Diamond System (BDS) (Section 2.6.2) was used to provide reference truth data.

Ideally, the experimental data set used to evaluate this filter design would be from a vehicle capable of highly dynamic motion. This is desirable because it would provide a wide range of real-world conditions for testing the dynamic range of the filter, and also because dynamic data provides the filter with more information about

its system states, leading to better performance. In this case, the only available vehicle with the necessary power supply and computing resources was a specially modified electric golf cart. While not ideal, the cart is dynamic enough to provide a data set that can give a basic evaluation of the design.

4.2.1 Data Collection. To collect the data, the BDS was mounted on the back of the cart, with its GPS antenna mounted approximately a meter above its IMU. Although it would have been ideal to have the antenna mounted directly on top of the IMU, the error introduced by the one meter vertical offset was considered small enough to be of negligible effect. The MT-9 was then mounted on top of the BDS IMU using a Velcro strip. With this setup, the BDS experienced the exact same dynamics as the MT-9 (within a range of error caused by the inexact mounting process). The BDS recorded its own data (time-stamped with GPS time) on an internal memory card, and a MATLAB script run on a laptop PC simultaneously recorded the MT-9 data. This MATLAB script recorded the raw accelerometer, gyroscope, magnetometer, and temperature readings from the MT-9, as well as the the internal MT-9 timer at each measurement interval. These readings were to be time-stamped with the GPS system time in order to correlate them with the data from the BDS. However, during data collection a software malfunction caused this time-stamp to be inaccurate. The method used to time-stamp the data correctly after the fact will be discussed shortly.

In order to provide several different data sets containing various dynamics, as well as one long data set, the MATLAB script that recorded the data was set to run for approximately fifteen minutes. During this time, the cart performed a series of right-hand circles, left-hand circles, and figure eights, as well a period of driving about randomly. This last period included starting and stopping frequently and forcefully, turning quickly in different directions, and riding up and down inclines to provide variation in pitch and roll. Between the various sets of maneuvers, the cart was stopped and remained still for thirty seconds or more. This was to provide the data

with several different “starting points”, because the Kalman filter software needed stationary data in order to initialize itself and perform self-alignment. The end result was a data set that could be broken down into the various sets of maneuvers, or used whole in order to provide a long-term analysis of the filter’s performance (this was the “long-term” data set).

4.2.2 Data Correction. Because of a software error in the MATLAB script, the MT-9’s data was not time-stamped correctly in real-time. In order to correct this and make the data usable, data from both the MT-9 and the BDS were examined. Along common axes, accelerometer data from both systems shared obvious peaks and valleys. When an identical peak or valley was found in both data sets, the BDS-recorded GPS time-stamp at this point was assigned to the MT-9 data at the same point. Once one MT-9 data point had a time-stamp, the MT-9’s internal timer data, which incremented once per millisecond, was used to time-stamp the rest of the data set accordingly. Once this was done, other unique peaks and valleys were used to compare the new MT-9 time-stamp to the BDS time-stamp: over the smallest data sets (usually around eighty seconds), the time-stamp variation was approximately 2 ms. Over the largest data set (13+ minutes), the maximum variation grew to roughly 11 ms. This variation is not ideal, but it was considered an acceptable error because of the relatively low dynamics of these tests (there were also significant technical problems in creating a replacement data set).

The Kalman filter also needs raw GPS pseudoranges and satellite ephemeris data, so the BDS-recorded GPS data was used. This means that some of the same raw data was used as an input to both the filter *and* the BDS (which provided the truth reference data). While this correlation between the filter data and the truth data is not ideal, the difference between two sets of single-frequency pseudoranges provided by different GPS receivers should be minimal (the ephemeris would be identical because it is directly downloaded from the satellites). Also, the techni-

cal complexity would have greatly increased with an attempt to record a separate, independent set of GPS measurements for the filter.

4.2.3 Data Processing. The computer that processed the input data and provided the filter output was a desktop PC with a Pentium 3 processor running at 1.4 GHz with 256 Megabytes of RAM. The largest data set (“long-term”), consisting of 100 Hz IMU data and 1 Hz GPS data over a 13.1 minute period, took an average (5-run average) of 48 seconds to process. This equates very roughly to a 16-1 speedup over real time. In a mobile, real-time implementation of the filter, a less powerful, much more space- and power-efficient processor would be used. The real-time code would also not just contain the INS mechanization and Kalman filter, but would also include data collection routines, time-stamp routines, and data output routines. However, the real-time code would also most likely be run on a stripped-down operating system to minimize other loads on the processor. All of these factors combine to make it very difficult to predict whether or not a small mobile processor can handle the filter’s computing requirements, although the high speedup of the desktop machine is an encouraging sign.

4.3 Filter Features

The system code has several important features that are not really relevant to Kalman filter theory or to the code structure (discussed in Chapters 2 and 3 respectively), so they will be discussed in this chapter. Figure 4.4 gives an overall system block diagram that shows some of the features to be discussed.

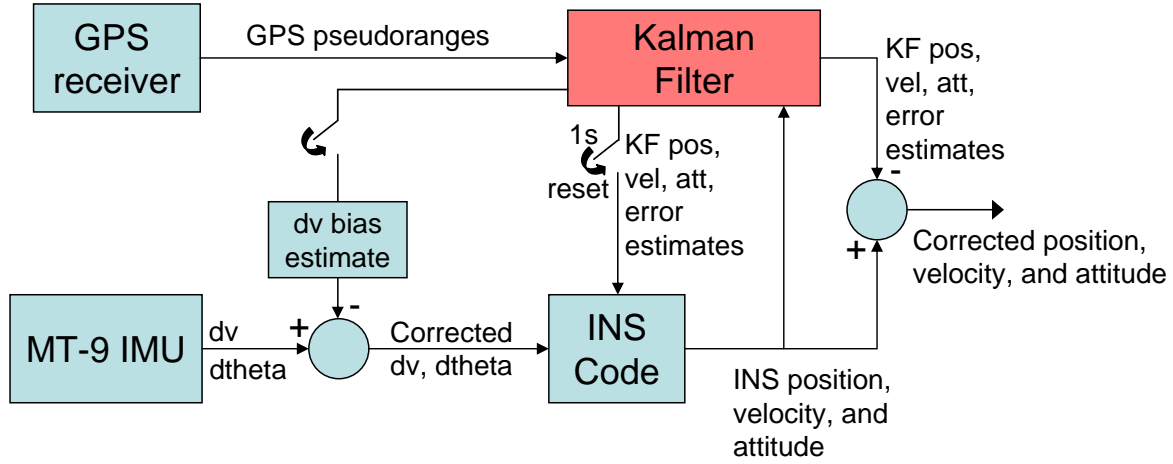


Figure 4.4 System Block Diagram

4.3.1 System Alignment. Before the Kalman filter begins estimating INS errors, the INS must first be aligned (that is, know its starting position and attitude). The starting position used for each data set was provided by the BDS truth position. This is realistic because in an actual real-time implementation the system's GPS receiver would be able to provide this starting position. However, a GPS receiver does not provide angular orientation, so for each data set the system had to calculate its own starting orientation.

The angular alignment calculations performed by the system are relatively simple. While this is partly because the focus of this research is on the system's performance *after* alignment, it was also somewhat of a necessity because the MT-9's instruments were not accurate enough to allow a very precise alignment. One important requirement of an accurate gyro-compassing INS alignment is that the gyroscopes sense the rotation rate of the earth, which is 0.00418 deg/s [10]. The MT-9's gyroscopes have a noise level of 0.745 deg/s [39], so it cannot accurately sense earth rate using a single measurement. While this noise level problem can be mitigated by averaging measurements over time, the bias instability that plagues MEMS gyroscopes (see Section 2.3.3) is the real problem. With the gyroscope bias unknown

from turn-on to turn-on, the earth rate can not be sensed even with averaging to remove the instrument noise.

A typical navigation-grade INS performs a “coarse” align, which provides a rough starting attitude, and then a “fine” align, which tightens up the alignment accuracy [30]. For this system, essentially a very coarse alignment is performed. During the first 5 seconds of data, the IMU must be sitting still (this is done to determine an initial gyroscope bias, as well as for alignment). During this time, all accelerometer, gyroscope, and magnetometer measurements in all three axes are recorded and average values are calculated (in order to reduce the effects of instrument noise). Because the INS is stationary, all sensed accelerations are due to gravity. With this knowledge, the initial roll and pitch determinations are easily calculated using the accelerometer averages. For the yaw angle, the MT-9’s magnetometers are used to sense the heading in relation to Earth’s magnetic field. This rough heading then has the magnetic variation (5.18 degrees at the starting position for the data sets [1]) subtracted from it, and the result is used as the initial heading for the INS. This subtraction is justifiable because the GPS receiver under consideration for a real-time implementation is the Garmin GPS-35, which, along with providing the system’s starting location, is capable of providing the magnetic variation for that location. Once the roll, pitch, and yaw angles are determined, a DCM is generated which relates the INS orientation to the navigation frame of reference.

The average error (using four different alignment data sets) for this angular alignment method, when compared to the BDS truth data, was 0.331 degrees in the roll axis and 0.105 degrees in the pitch axis. The yaw axis, which is the most difficult to align (because of the lack of gravity in this axis), had an average error of 4.62 degrees.

4.3.2 Instrument Error Feedback. In the system code there are separate variables that are subtracted from every gyroscope and accelerometer measurement.

These variables represent the estimated instrument biases. During the five-second stationary alignment period, the gyroscope measurements are summed and an average value is formed (one for each gyroscope). With perfect instruments, these values would be zero, because the IMU is not rotating (other than the very small earth rate rotation). Because of instrument noise and bias, with bias being the greatest contributing factor, these values are not zero. They are placed into the variables that are subtracted from every gyroscope measurement.

Accelerometers cannot use the same process to identify their biases (unless the attitude of the IMU is known accurately), because gravity causes them to sense accelerations even when the IMU is static. Their bias variables are therefore set to zero initially. Once the filter begins operation, and its estimate of any of the three accelerometer biases grows above 0.08 m/s^2 , these bias estimates are inserted into the variables that are subtracted from every accelerometer measurement. The filter's accelerometer bias estimates are then reset to zero, and this reset process continues the whole time the filter is running. The gyroscope bias states are not fed back in this manner. The justification for not feeding back the gyroscope biases can be seen in Figure 4.5, which shows filter estimates of the Z-axis accelerometer and gyroscope biases from data set "long-term". The accelerometer bias state adjusts quickly from its value of zero, and remains fairly constant once it settles on a value. This seems to indicate that the filter has settled on a reasonable value for the accelerometer bias. In contrast, the gyroscope error state wanders continuously, giving an indication that a solid bias value has not really been discovered (assuming that the true gyroscope bias is not as fast-changing as the filter is indicating).

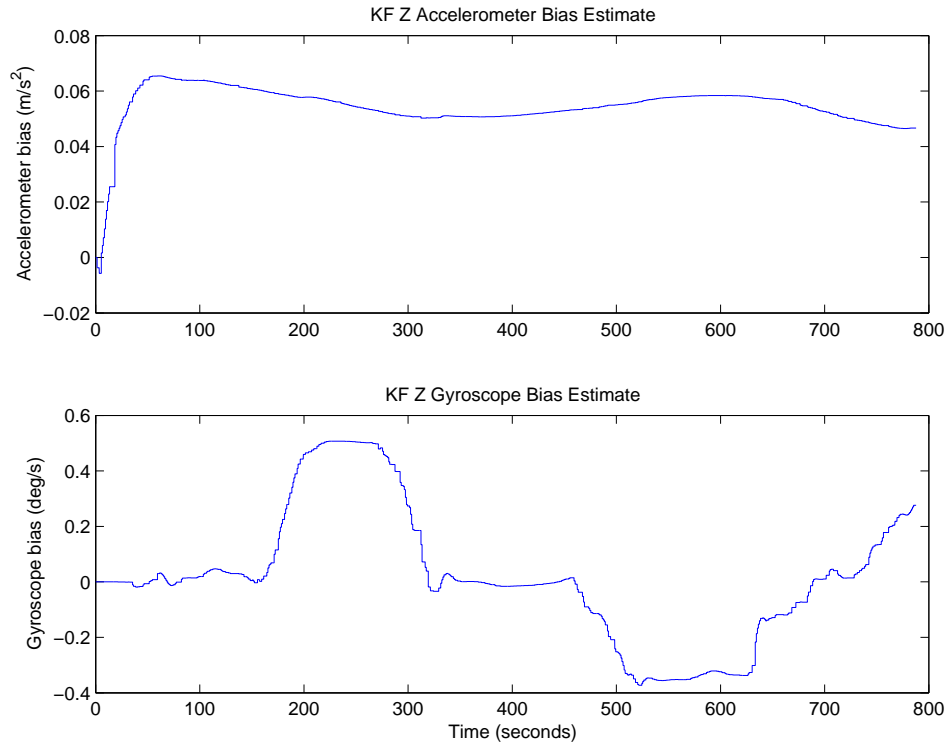


Figure 4.5 Filter Estimates of Z-axis Accelerometer and Gyroscope Biases, “long-term”

This feedback process is implemented in order to provide more accurate measurements to the filter’s INS mechanization, which will help its error to grow more slowly. A very obvious effect of this feedback is shown by Figure 4.6, which shows two sets of data, one with accelerometer bias feedback and one without (this data is from the “Figure Eights” data set). The BDS-recorded down velocity is used as the truth data (black line), and the MT-9’s indicated down velocity (green or gray line) ideally would track this truth data. It can be seen that about 25 seconds into the “feedback” data, the MT-9 suddenly stops its drift away from the truth down velocity. From this point on, the “feedback” MT-9 down velocity tracks the truth data much closer than the non-feedback data. This is an indication that a bias state reset occurred, with the result that the MT-9’s error growth was slowed.

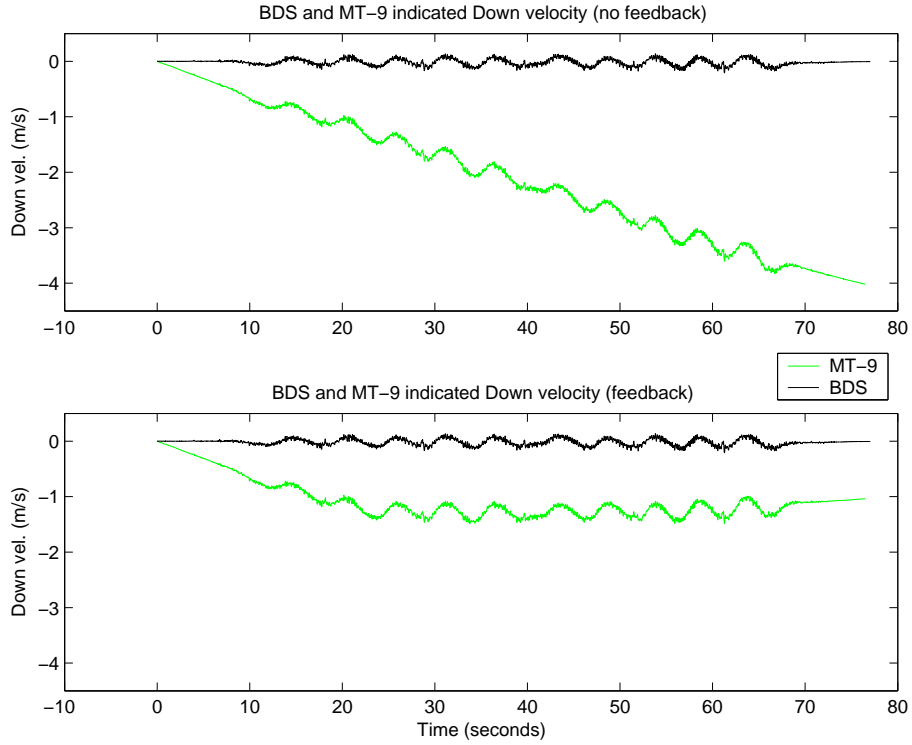


Figure 4.6 MT-9 and BDS Indicated Down Velocities, with and without Feedback, “figure eights”

4.3.3 INS Resets. As time progresses, the INS position, velocity, and attitude outputs drift away from the truth. As these errors grow larger, it becomes more and more difficult for the Kalman filter to estimate them correctly. One reason is because a large INS error violates a key assumption of the filter measurement model (see the explanation for Figure 3.1). To address this problem, the INS is reset after every filter cycle, whether it is a state propagate/update cycle or just a propagate cycle. A reset is when the INS mechanization essentially starts over, using as new starting values its previous outputs corrected by the Kalman filter’s estimated errors. These filter error estimates are then reset to zero (all other filter states remain as they were before). Initially, the INS was going to be reset whenever the filter’s error estimates grew above certain bounds, but it was found that, as the INS was reset more and more frequently, the overall system accuracy increased.

Figure 4.7 shows a simple comparison between the filter with and without INS resets implemented (both using the same data set, “long-term”). The results are clear; with the INS resets, the filter’s indicated north position tracks right along with the BDS truth data, whereas without them the filter’s indicated north position is off by 50 kilometers by the end of the data set.

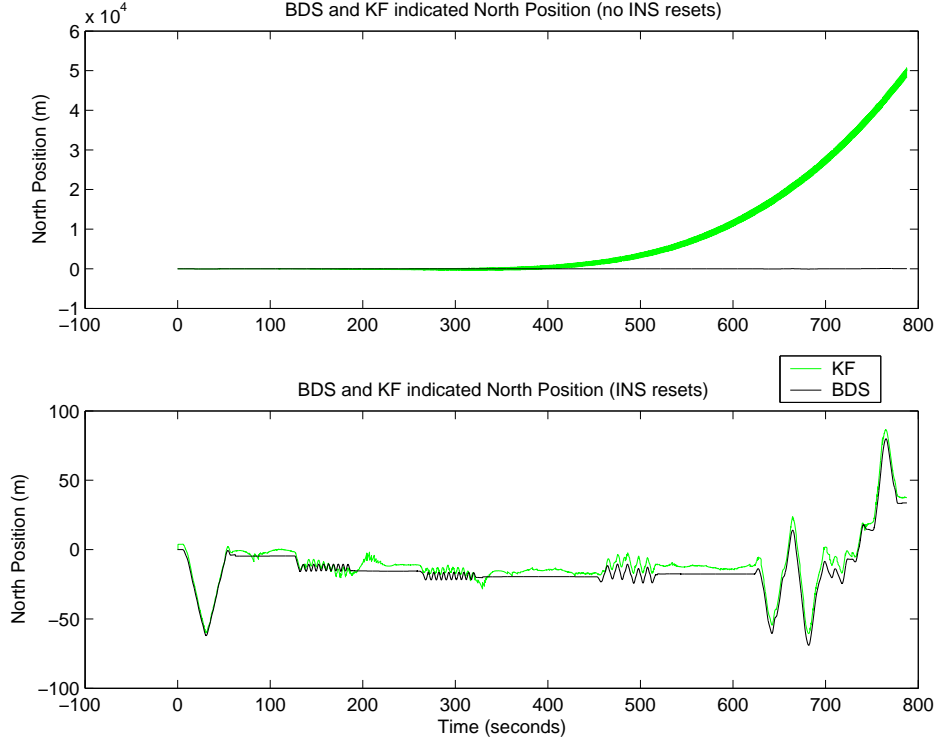


Figure 4.7 North Position Filter Performance, with and without Feedback, “long-term”

While the Kalman filter performs its position calculations for latitude and longitude using decimal degrees, the plots of these errors make much more sense when meters are used as the scale. Therefore, all of the position plots seen in this chapter (such as Figure 4.7) show north and east position in meters. To convert latitude and longitude in degrees to north and east in meters, the filter outputs were multiplied by scale factors before plotting. The scale factors are calculated as shown in the following equations, where L is the latitude and R_e is the equatorial radius of

the earth (6,378,137 meters [10]).

$$Latitude\,factor = \frac{R_e\pi}{180} \quad (4.1)$$

$$Longitude\,factor = R_e \cos\left(\frac{L\pi}{180}\right) \frac{\pi}{180} \quad (4.2)$$

A final note on the plots is that altitude is referenced in meters above the WGS-84 ellipsoid.

4.4 *Filter Tuning*

4.4.1 Initial State Covariances. The Kalman filter's initial state variances are shown in Table 4.2. They are shown as squared standard deviations. Practically, these numbers tell the filter that the true initial error states have a 68.3 percent probability of being within (plus or minus) these standard deviation values.

Table 4.2 Initial Filter State Covariances

Filter State	Initial State Covariance
δlat	$(10 \text{ m})^2$
δlon	$(10 \text{ m})^2$
δalt	$(10 \text{ m})^2$
δV_N	$(1 \times 10^{-10} \text{ m/s})^2$
δV_E	$(1 \times 10^{-10} \text{ m/s})^2$
δV_D	$(1 \times 10^{-10} \text{ m/s})^2$
$\delta \alpha$	$(0.4 \text{ deg/s})^2$
$\delta \beta$	$(0.4 \text{ deg/s})^2$
$\delta \gamma$	$(4.0 \text{ deg/s})^2$
δt	$(20 \text{ m})^2$
$\delta \dot{t}$	$(0.1 \text{ m/s})^2$
$bias_{xAccel}$	$(0.5 \text{ m/s})^2$
$bias_{yAccel}$	$(0.5 \text{ m/s}^2)^2$
$bias_{zAccel}$	$(0.5 \text{ m/s}^2)^2$
$bias_{xGyro}$	$(0.1 \text{ deg/s})^2$
$bias_{yGyro}$	$(0.1 \text{ deg/s})^2$
$bias_{zGyro}$	$(0.1 \text{ deg/s})^2$

The initial position standard deviations are 10 meters because this is a reasonable accuracy for a GPS position solution, which is used to provide the INS starting point. The velocity standard deviations are zero for all practical terms, because of the extreme confidence that the system was not moving when initialized. The pitch, roll, and yaw standard deviations reflect the accuracy of the INS alignment routine (see the end of Section 4.3.1). The time state and accelerometer bias state standard deviations are somewhat high because of uncertainty in these values at the start. The gyroscope bias standard deviations are relatively low, because during the align-

ment phase a preliminary value was already calculated, giving the filter a reasonably accurate start.

4.4.2 Noise Values. The Kalman filter's system dynamics noise values are shown in Table 4.3. They are shown as squared standard deviations. These values were chosen after much time spent tuning the filter.

Table 4.3 System Dynamics Noise (**Q** matrix)

Filter State	Noise Value
δlat	$(6.286 \times 10^{-10} \text{ m})^2 \text{sec}$
δlon	$(6.286 \times 10^{-10} \text{ m})^2 \text{sec}$
δalt	$(6 \text{ m})^2 \text{sec}$
δV_N	$(1 \times 10^{-6} \text{ m/s})^2 \text{sec}$
δV_E	$(1 \times 10^{-6} \text{ m/s})^2 \text{sec}$
δV_D	$(1 \times 10^{-6} \text{ m/s})^2 \text{sec}$
$\delta \alpha$	$(2 \times 10^{-9} \text{ deg})^2 \text{sec}$
$\delta \beta$	$(2 \times 10^{-9} \text{ deg})^2 \text{sec}$
$\delta \gamma$	$(2 \times 10^{-9} \text{ deg})^2 \text{sec}$
δt	$(1.0 \text{ m})^2 \text{sec}$
$\delta \dot{t}$	$(0.2 \text{ m/s})^2 \text{sec}$
$bias_{xAccel}$	$(8 \times 10^{-4} \text{ m/s}^2)^2 \text{sec}$
$bias_{yAccel}$	$(8 \times 10^{-4} \text{ m/s}^2)^2 \text{sec}$
$bias_{zAccel}$	$(8 \times 10^{-4} \text{ m/s}^2)^2 \text{sec}$
$bias_{xGyro}$	$(0.01 \text{ deg/s})^2 \text{sec}$
$bias_{yGyro}$	$(0.01 \text{ deg/s})^2 \text{sec}$
$bias_{zGyro}$	$(0.01 \text{ deg/s})^2 \text{sec}$

These filter noise values affect how much error the filter assigns to each state as it propagates forward in time. They also heavily influence the filter covariances,

which describe how confident the filter is in each state estimate. Plots comparing the filter’s error estimates and covariances to the true errors are shown later in this chapter.

The filter’s measurement noise variance value controls how much the filter “trusts” the measurements it receives. This value was set at $(7 \text{ meters})^2$, a reasonable uncertainty for the measurement model described in Section 3.2.3.

4.5 Test Results

In this section, the filter’s performance is examined over a wide range of scenarios. Short-term accuracy and long-term accuracy is examined, as well as performance during a significant GPS outage.

4.5.1 Short-term Performance. The data set chosen to analyze the filter’s short-term performance was “short-term”, referenced in the chapter overview. Figures 4.8 through 4.11 show comparisons between the BDS data and the filter outputs. The BDS indicated truth for each state are the black lines, and the filter’s indicated values are the green or gray lines. Figures 4.12 through 4.14 compare the filter’s error to its covariance-derived standard deviation estimates, with the error calculated by subtracting the filter’s results from the BDS results. For these error plots, the error is black and the filter’s standard deviations are green or gray. Table 4.4 summarizes the plotted data. Truth data is only available for the first nine states, so those states are what is shown (see Section 4.5.2.1 for plots and analysis of the other filter states).

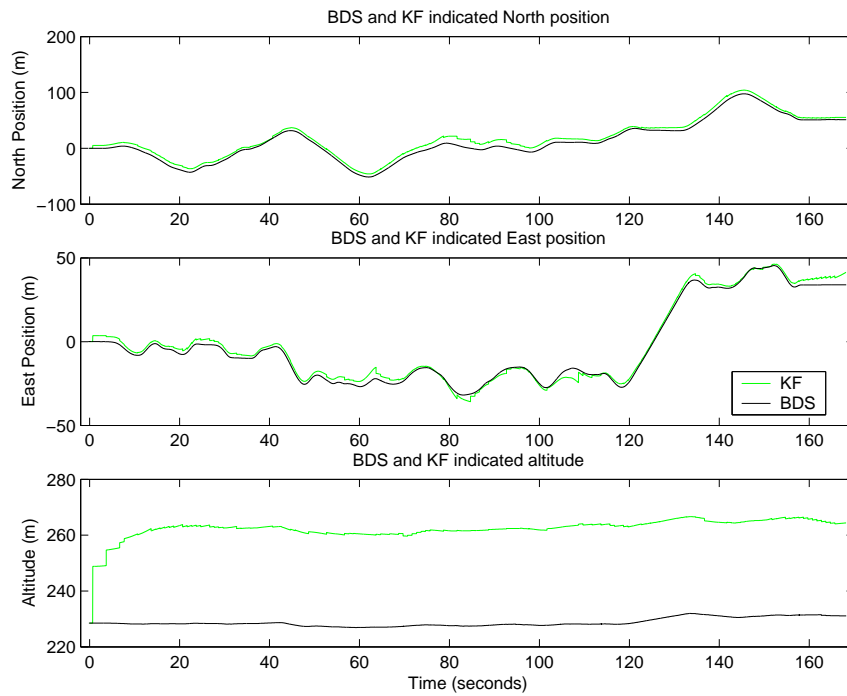


Figure 4.8 North, East, and Altitude Filter Performance, “short-term”

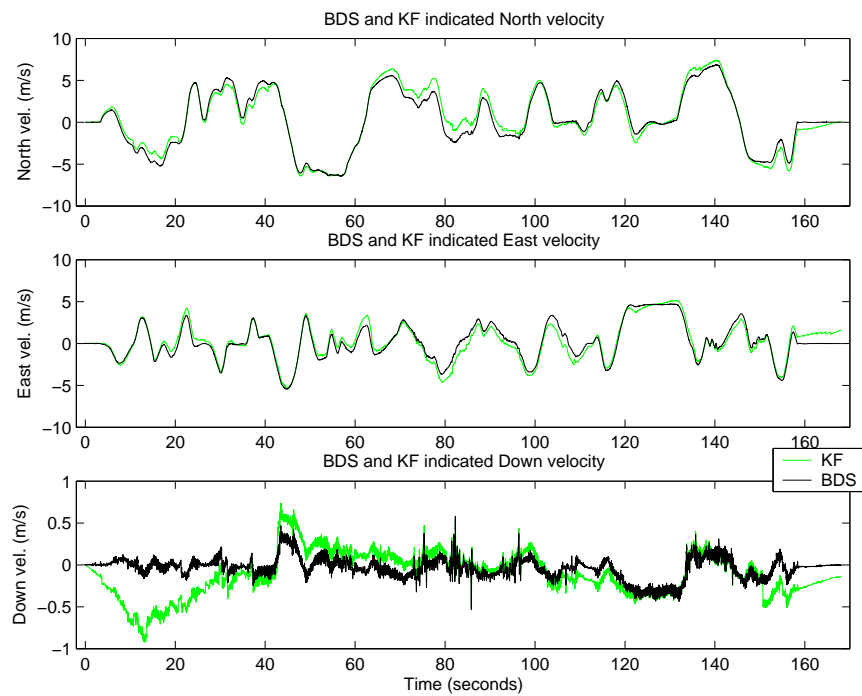


Figure 4.9 North, East, and Down Velocity Filter Performance, “short-term”

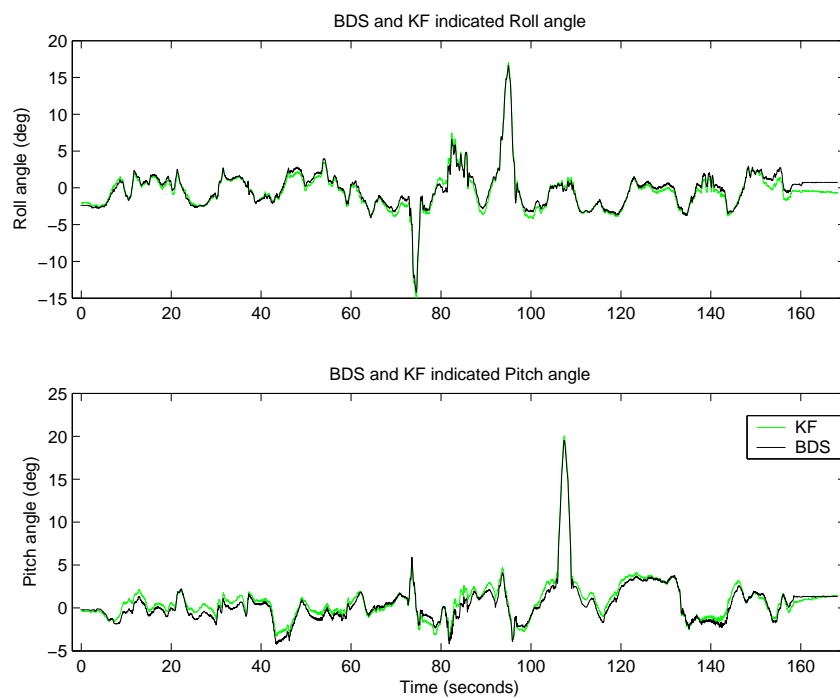


Figure 4.10 Roll and Pitch Filter Performance, “short-term”

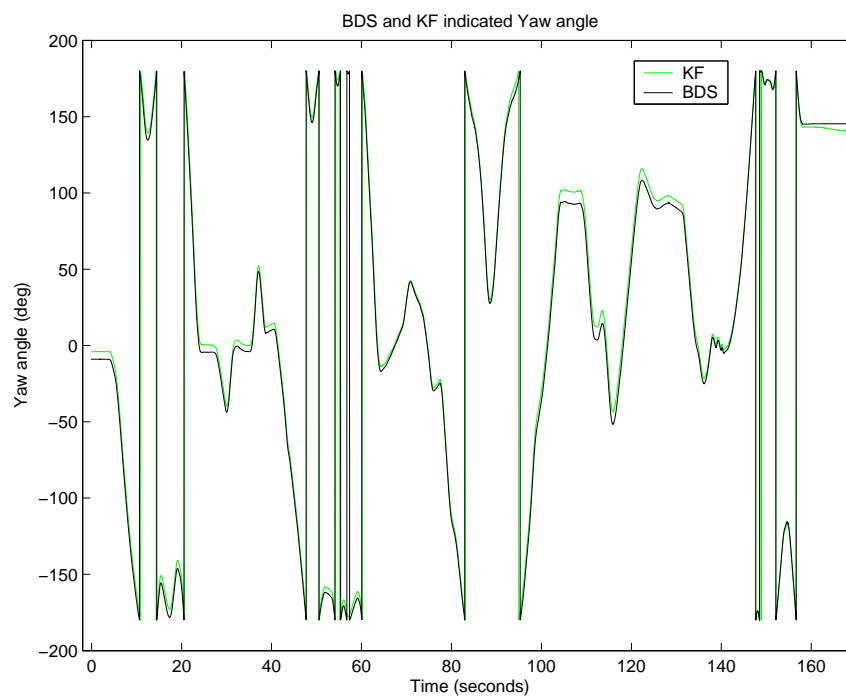


Figure 4.11 Yaw Filter Performance, “short-term”

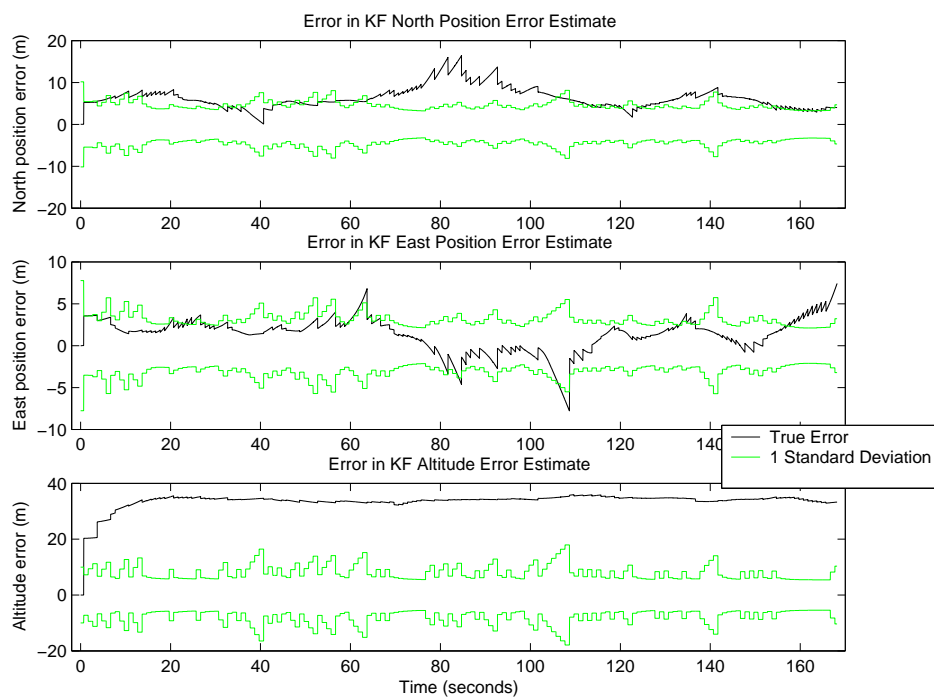


Figure 4.12 Position Errors and Standard Deviations, “short-term”

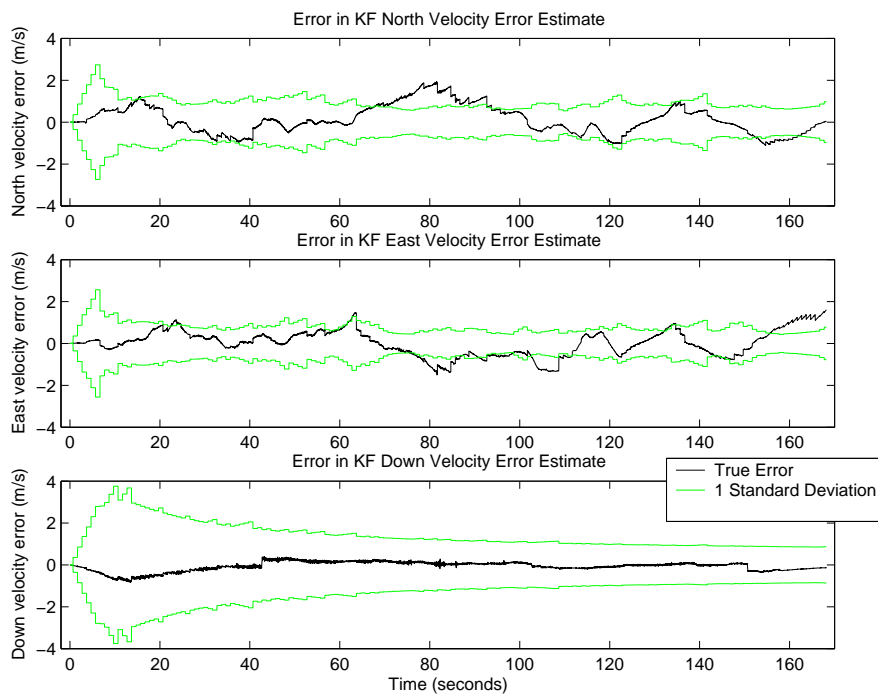


Figure 4.13 Velocity Errors and Standard Deviations, “short-term”

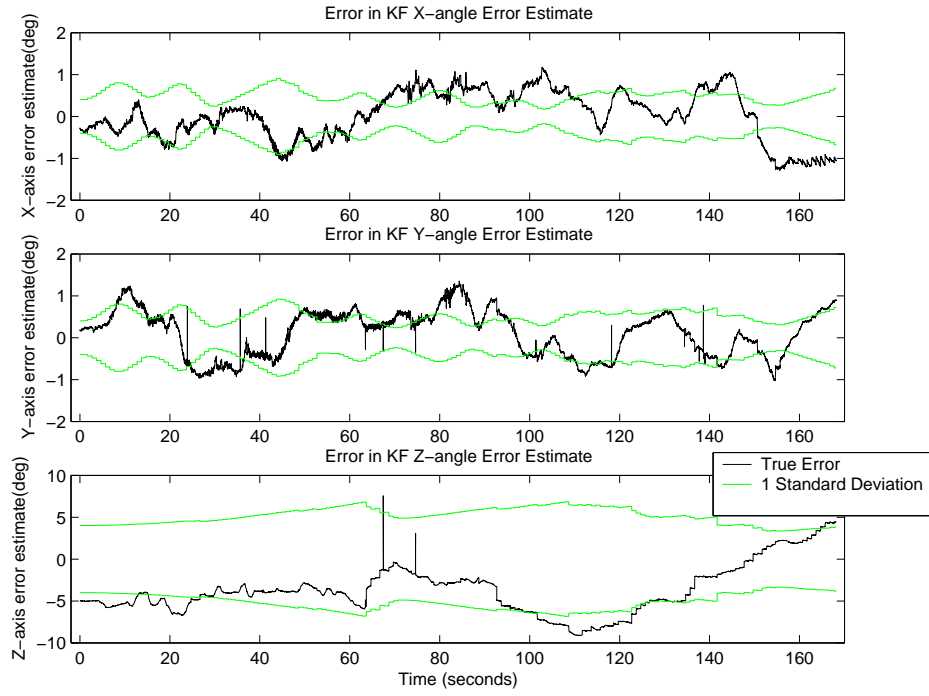


Figure 4.14 X, Y, and Z-Axis Tilt Errors and Standard Deviations, “short-term”

Table 4.4 Short Term Performance Errors

Filter Output	Mean Error	Error Std. Dev.	Pct. within 1σ
North Position	6.31 m	2.56 m	23.1
East Position	1.32 m	2.01 m	82.5
Altitude	33.6 m	3.13 m	0.43
North Velocity	0.126 m/s	0.678 m/s	78.3
East Velocity	0.0225 m/s	0.621 m/s	70.4
Down Velocity	-0.0718 m/s	0.225 m/s	99.6
X-tilt	0.0230 deg	0.585 deg	53.6
Y-tilt	0.103 deg	0.546 deg	54.3
Z-tilt	-3.70 deg	3.17 deg	67.9

In Figure 4.8, a very pronounced altitude bias is visible. This is further confirmed in Table 4.4, as the average altitude error is 33.6 meters. Thinking that such

a bias would most likely be introduced by faulty GPS processing, the output of the system’s GPS code was double-checked against output from verified MATLAB code (both using the same inputs), and the positions were all the same. Therefore, this altitude bias is most likely due to some extra GPS processing the BDS does that the system code does not replicate. Dual-frequency pseudoranges or a more elaborate troposphere or ionosphere correction model could be the source of such a discrepancy (the MEMS-based system does use a tropospheric correction model, a modified Hopfield model [13]). Aside from this altitude error, the rest of the error statistics seem reasonable. Most of the errors are at least somewhat close to zero-mean, with the north position, altitude, and z-tilt errors showing the largest error means. The standard deviations all show reasonably tight error distributions.

The filter’s standard deviations are the measure of the filter’s confidence in its outputs. Kalman filter theory says that, with a well-tuned filter, the filter’s outputs should be within one standard deviation of the truth 68.3 percent of the time [20]. Most of the filter outputs (see the fourth column of Figure 4.4) are reasonably close to this number, with the exception of the north, altitude, and down velocity errors. The altitude errors have already been discussed. The north errors, seen in Figure 4.12, are all very close to one standard deviation, even if a majority of them fall outside this boundary. The down velocity error, seen in Figure 4.13, is very small. Attempts to reduce the system dynamics noise for this state resulted in little or no change to the filter covariances. Basically, with other tuning parameters remaining the same, the filter is as confident as it will get when it comes to the down velocity state. This test was successful in showing that the filter’s performance is adequate over a short time. Each error state was roughly constant over the entire data set, which is what would be expected over a short period of time. However, the true test of the system’s performance comes when a longer data set is examined.

4.5.2 Long-term Performance. The long-term performance of the filter was examined using data set “long-term” (13.1 minutes long). Figures 4.15 through

4.18 show comparisons between the BDS data and the filter outputs. Figures 4.19 through 4.21 compare the filter’s error to its covariance estimates, and Table 4.5 summarizes the data. Again, truth data is only available for the first nine states.

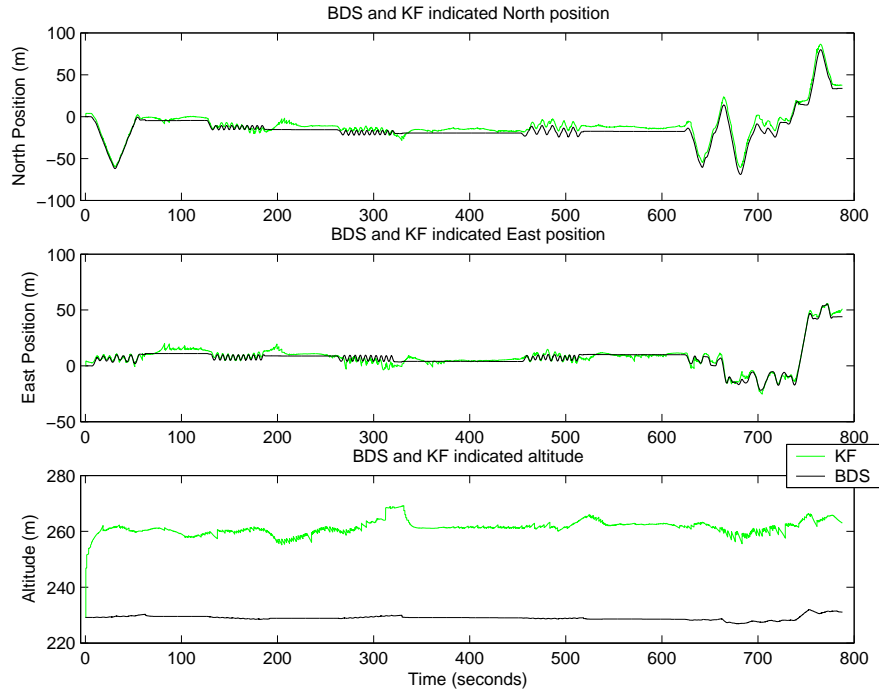


Figure 4.15 North, East, and Altitude Filter Performance, “long-term”

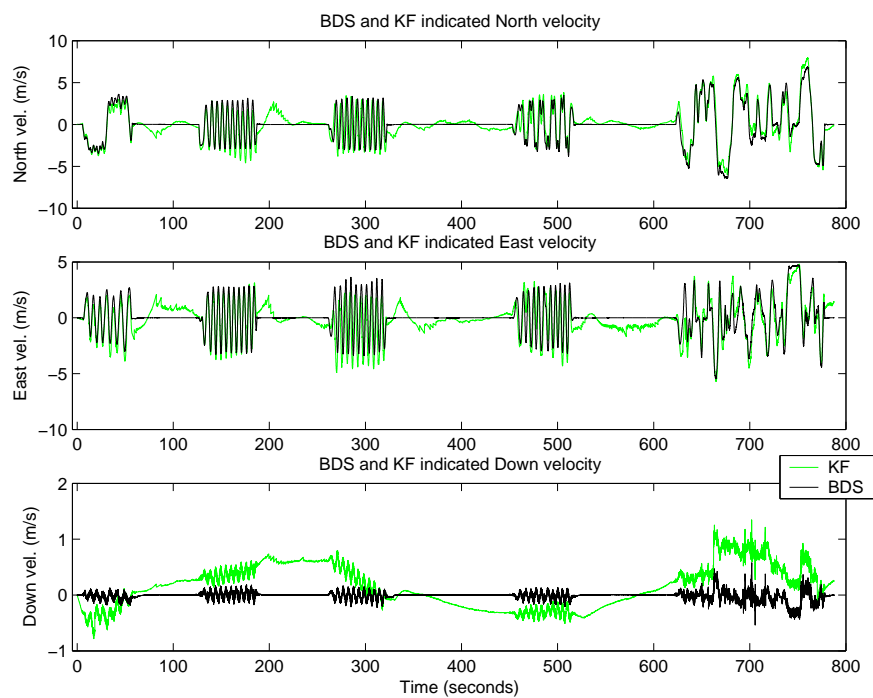


Figure 4.16 North, East, and Down Velocity Filter Performance, “long-term”

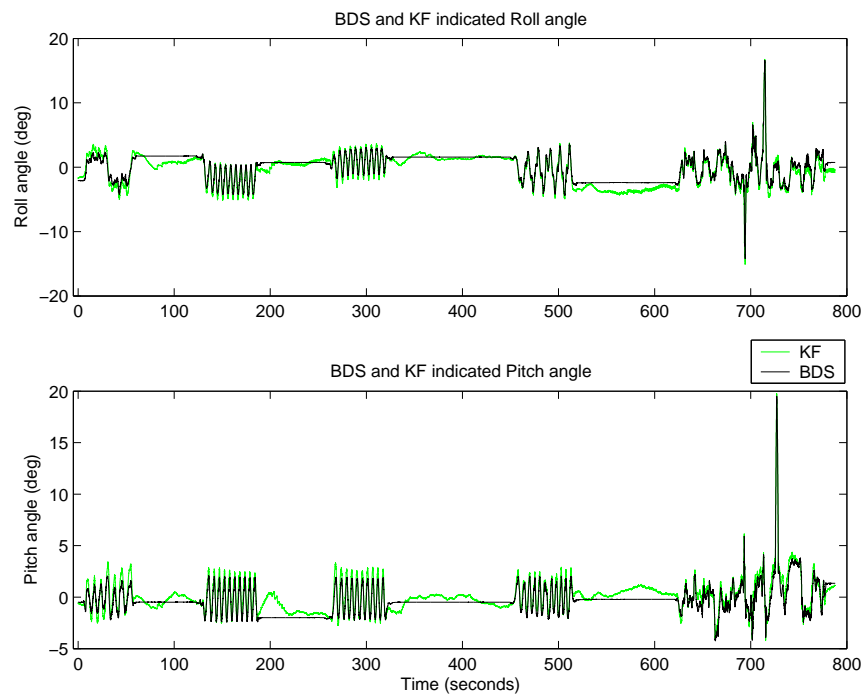


Figure 4.17 Roll and Pitch Filter Performance, “long-term”

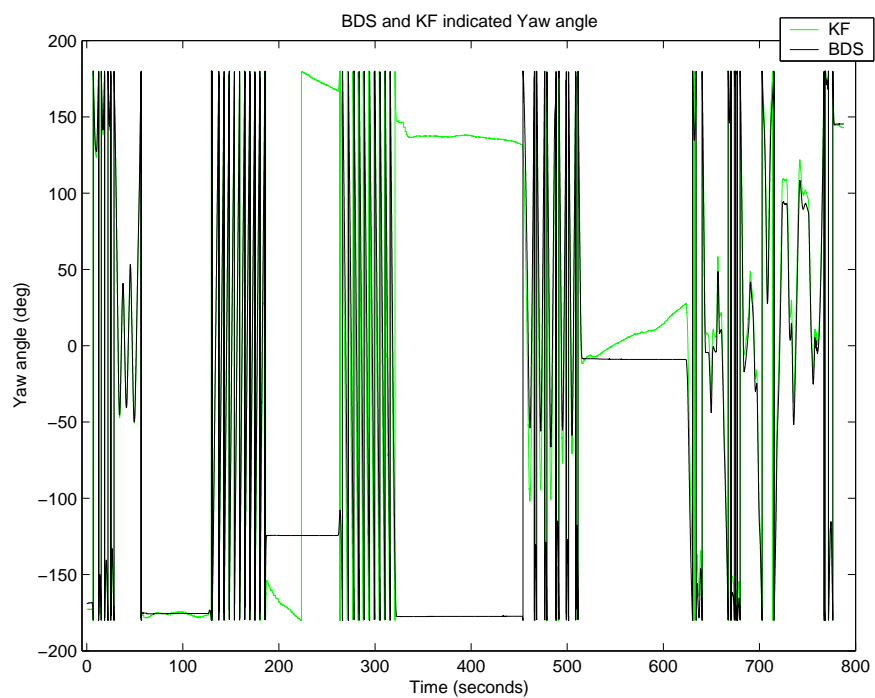


Figure 4.18 Yaw Filter Performance, “long-term”

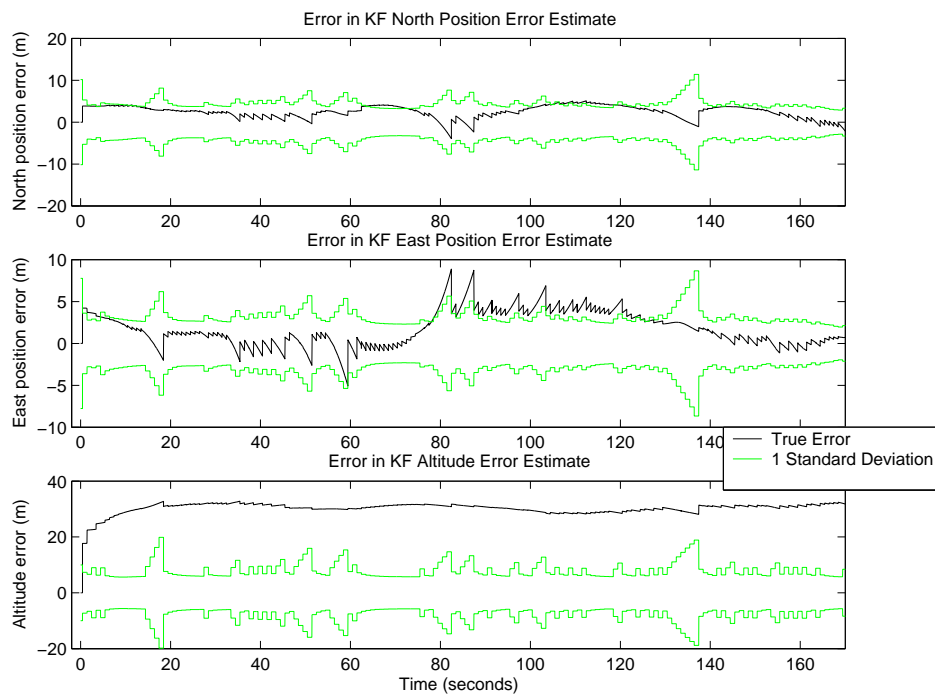


Figure 4.19 Position Errors and Standard Deviations, “long-term”

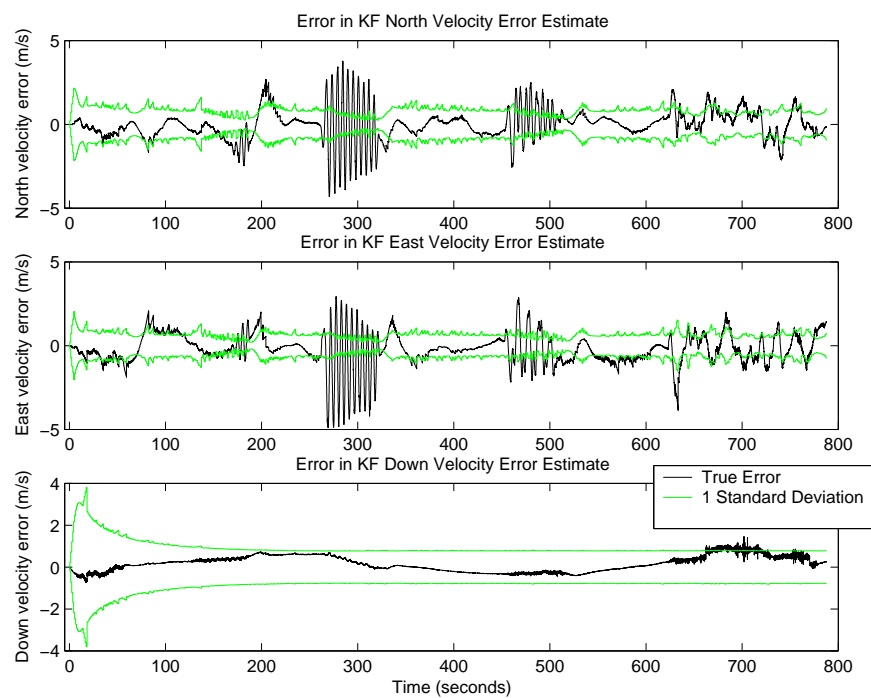


Figure 4.20 Velocity Errors and Standard Deviations, “long-term”

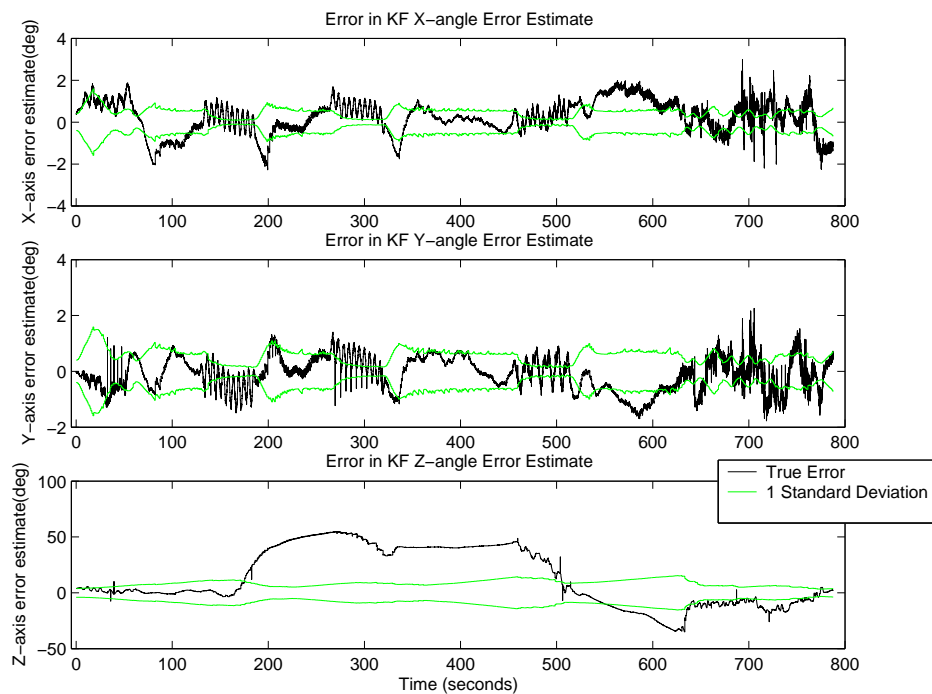


Figure 4.21 X, Y, and Z-Axis Tilt Errors and Standard Deviations, “long-term”

Table 4.5 Long Term Performance Errors

Filter Output	Mean Error	Error Std. Dev.	Pct. within 1σ
North Position	4.33 m	3.01 m	46.5
East Position	0.752 m	2.53 m	75.3
Altitude	32.3 m	2.36 m	0.057
North Velocity	-0.0174 m/s	0.923 m/s	73.4
East Velocity	-0.229 m/s	0.986 m/s	52.7
Down Velocity	0.162 m/s	0.371 m/s	94.4
X-tilt	0.237 deg	0.786 deg	38.8
Y-tilt	-0.0999 deg	0.604 deg	60.6
Z-tilt	13.3 deg	25.4 deg	30.3

The filter statistics contained in Table 4.5 are very similar to those from data set “short-term”. Most error means did not change significantly, though the standard deviations for the velocity and X and Y tilt states all grew slightly. The X and Y tilt error state percentages are also somewhat low, considering that the same tuning values were used for data set “short-term” with better results. However, it is not a good idea to tune the filter just to optimize one data set. The most significant change is the mean error and error standard deviation for the MT-9’s Z-axis tilt, which are 13.3 and 25.4 degrees respectively. The main reason for these large errors is that this data set had a lot of time during which the golf cart was static. In Figure 4.18 (especially between 500 and 600 seconds) it is apparent that the yaw error growth is the most rapid during these static periods, and the filter takes quite a while to correct this error. Figure 4.21 demonstrates that during static periods, the filter’s angular covariances grow, indicating less confidence in its estimates. While the covariances grew enough to account for the X and Y-tilt errors, the Z-tilt covariances did not grow enough to account for the large error growth in that axis. This is partially because the filter is tuned more for dynamic data than stationary data, but

the covariance numbers in this axis could probably be improved with better filter tuning. Another way to improve them would be to add GPS velocity as another filter measurement (see the next paragraph for an explanation). During motion, the covariances shrink, because motion allows the filter to discriminate between various error sources, and thus more accurately estimate the appropriate states. Figure 4.22, which is a zoomed-in version of Figure 4.18, shows a graphic example of this. It shows that the filter’s estimate of the yaw gets closer and closer to the truth as the cart’s motion continues. The inability of the MT-9 to sense earth rate, aggravated by the lack of motion, causes the large yaw error growth seen during the static periods.

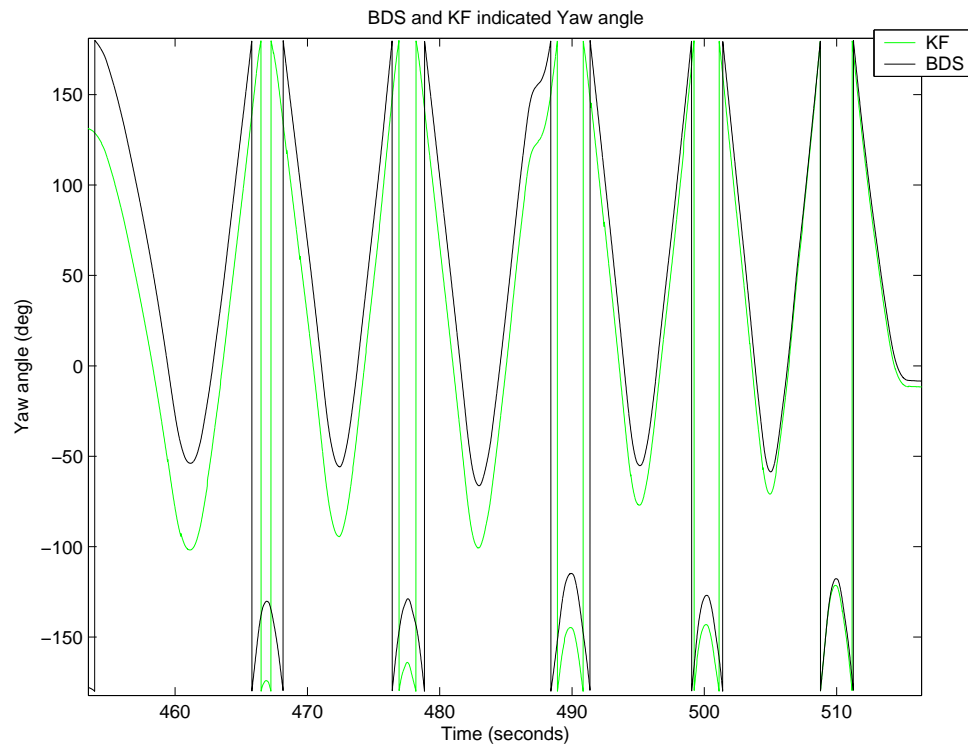


Figure 4.22 Zoomed-in Yaw Filter Performance, “long-term”

It is also interesting to note Figure 4.20, which shows the error in the north and east velocity error estimates. At the 300-second point, these two velocity errors are oscillating wildly, with error magnitudes of nearly 5 meters/second (this period of motion was the golf cart going continuously in tight circles). The filter covariances

do not match the errors correctly, as they are much too optimistically low. The probable source of this error can be seen in Figure 4.21 in the Z-axis tilt error. At the 300-second mark, the filter’s heading estimate is off by close to 50 degrees. This causes the filter’s north and east axes to be misaligned with the true north and east axes, causing the large velocity error oscillations. Another interesting thing to note is that, although the cart is in motion at this time, the filter does not recognize that there is a large heading error; in fact, the heading error is just as big at the end of the motion as it was at the beginning. Only when the cart starts moving again (around 460 seconds, the “figure eights” data set), does the filter realize the heading is off and start assigning error to this state. A possible explanation for this is that the cart’s circles at the 300-second point were small enough to be within the GPS margin of error, while the “figure eights” data set contained enough change in position for the filter to realize that it had a large heading error. One way that this problem would have been avoided was if the filter used GPS velocity measurements. During the cart’s tight circles, the GPS velocity would have allowed the filter to realize its errors and correct them. The GPS position alone was not enough to do this when the position change during motion was so small.

The results of this long-term data set lead to some guardedly optimistic conclusions. The filter performed well during dynamic conditions, and showed an ability to correct any heading errors that appeared during static periods. However, this heading drift was quite large at certain points and further tuning should be attempted to try to correct this problem.

4.5.2.1 Filter Clock and Bias States. Although no truth data is available for the last eight filter states, they are still important to understanding the filter’s performance. It was clear during the tuning process that the bias noises in particular were critical to filter performance, as they provided the greatest improvements when changes were made. Figures 4.23 through 4.28 show these states and their covariance-derived standard deviations. It is important to remember that

these states do not show the error in the estimates as the other plots do, but just the filter estimates themselves (the error estimates and standard deviations have been plotted separately because of this). The standard deviations show the filter’s confidence level, but there is no way of knowing actually how good these estimates are.

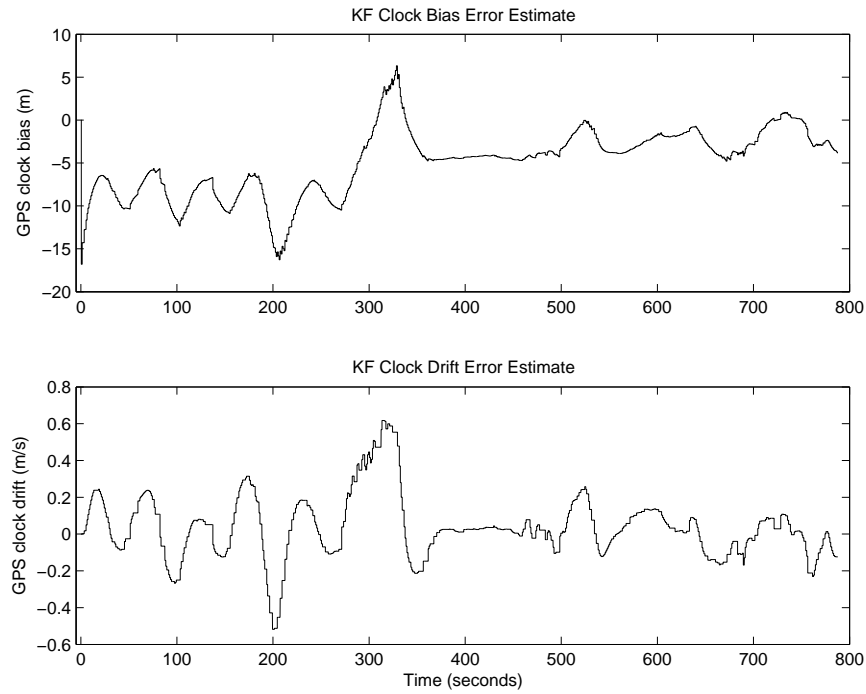


Figure 4.23 Clock Bias and Clock Bias Drift Estimates, “long-term”

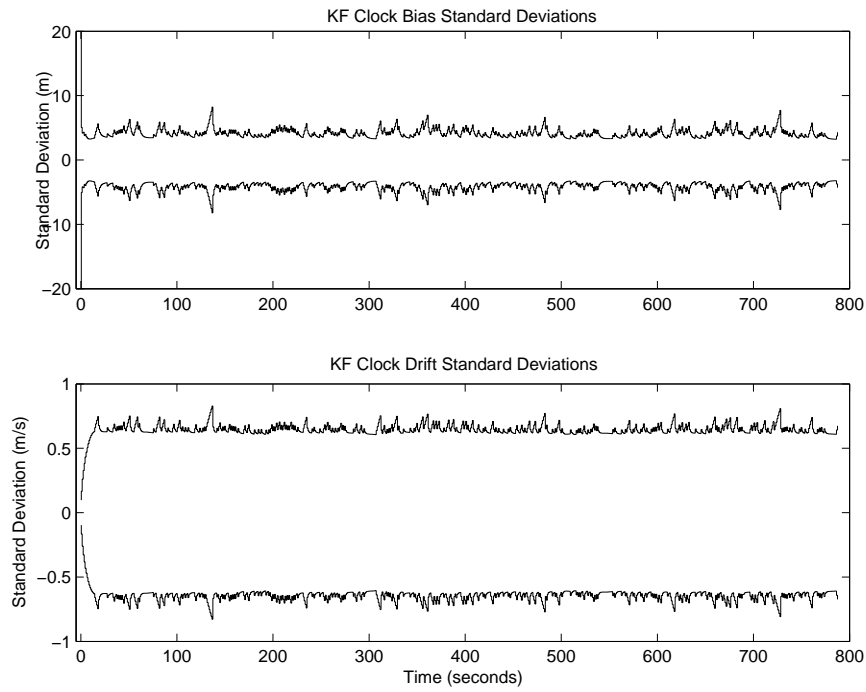


Figure 4.24 Clock Bias and Clock Bias Drift Standard Deviations, “long-term”

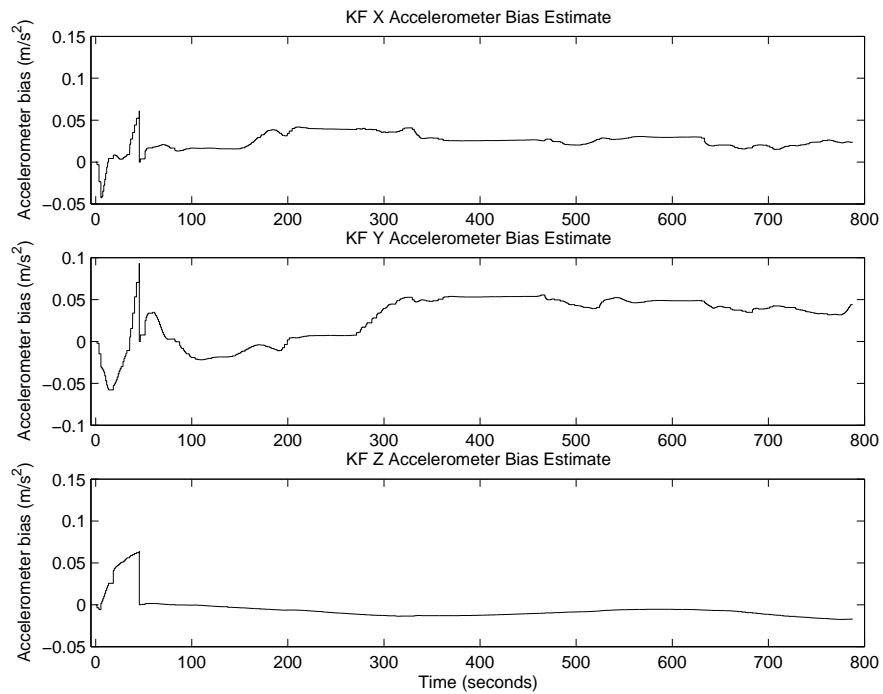


Figure 4.25 X, Y, and Z Accelerometer Bias Estimates, “long-term”

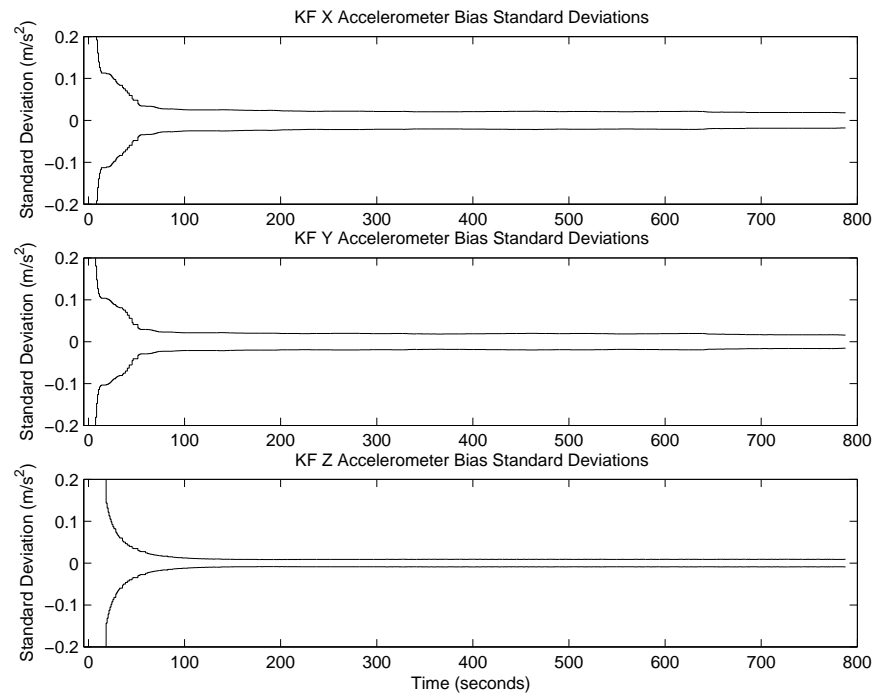


Figure 4.26 X, Y, and Z Accelerometer Bias Standard Deviations, “long-term”

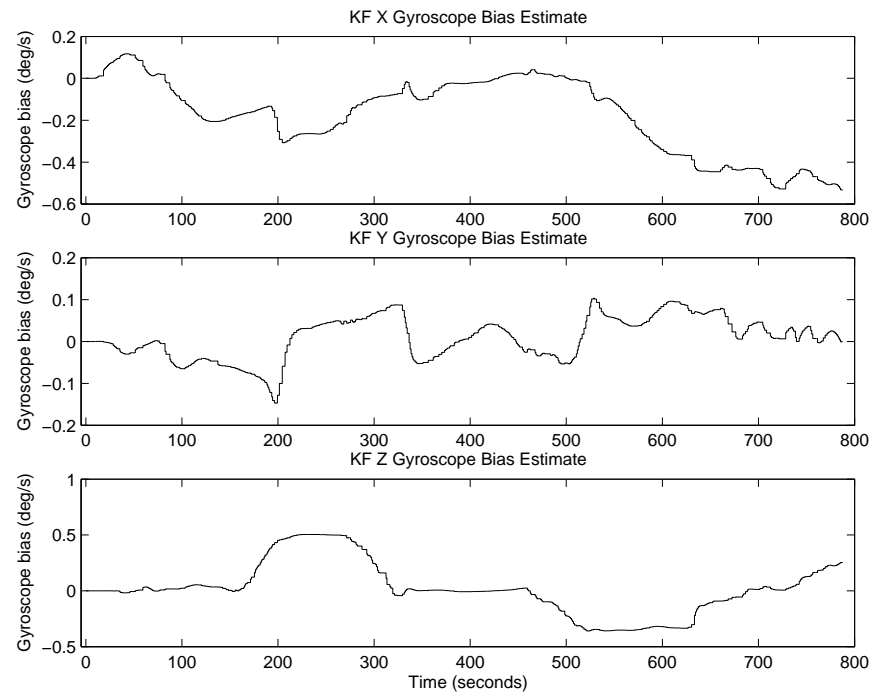


Figure 4.27 X, Y, and Z Gyroscope Bias Estimates, “long-term”

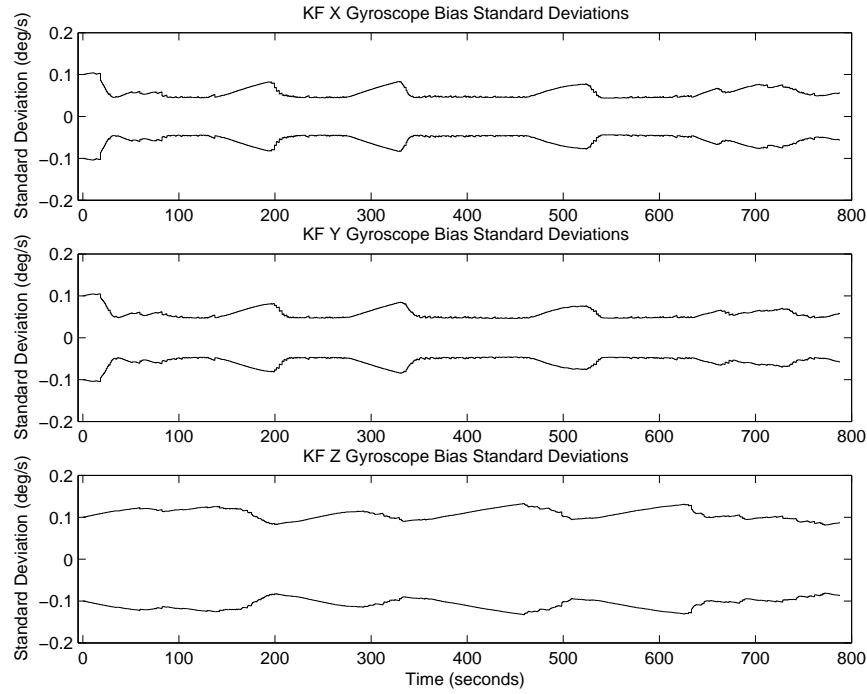


Figure 4.28 X, Y, and Z Gyroscope Bias Standard Deviations, “long-term”

Both the clock bias and bias drift states appear to be reasonable, although there is no way to verify them. The accelerometer biases undergo a reset at about the 40-second point, just as their covariances are converging from their initial values. After this reset they stay very steady, which is an encouraging sign that these states have a good connection to reality. On the other hand, the gyroscope biases wander and never settle on steady values. One possible reason for this behavior is that the noise for these states could be too high, causing the filter to assign too much error to these states. Better tuning would be the answer if this was the case. Another reason could be that there are other types of gyroscope error that are causing problems, such as scale factor error. In this case, more detail would need to be added to the filter’s dynamics model.

One point of interest is the Z gyroscope bias from about 270 to 320 seconds. This is the period of “tight circles” discussed in the previous section, when the filter

failed to correct the heading error during motion. From viewing the Z gyroscope bias, it appears that the filter was aware of a problem, and was rapidly lowering its Z gyroscope bias estimate to try to fix it, to no avail. This possibly points to the conclusion that better tuning could fix this problem. If the relationship between the angular state noise and the gyroscope bias state noise was different, perhaps the filter would have assigned the error to the heading error state, instead of improperly assigning it to the Z-axis gyroscope bias state as it did here.

This long data set was a serious test for this MEMS-based system. The success of the filter at maintaining a reasonable output proves that an inexpensive MEMS-based solution is a viable option for short to medium-term accuracy. However, the filter's accuracy is best when the system is moving. A long static period could possibly result in the yaw output becoming unstable, with the filter being unable to correct itself.

4.5.3 GPS Outage. To analyze the filter's performance with a GPS outage, data set "short-term" was again used, but with GPS data removed so that there was a 31-second gap where no pseudoranges were available. The filter's north and east direction and velocity states show the greatest effects of the outage, so these are the states discussed in this section. Figures 4.29 through 4.32 show the performance of these states, and Table 4.6 shows the filter's error statistics. The period of GPS outage is indicated approximately by the two vertical black lines in all four plots.

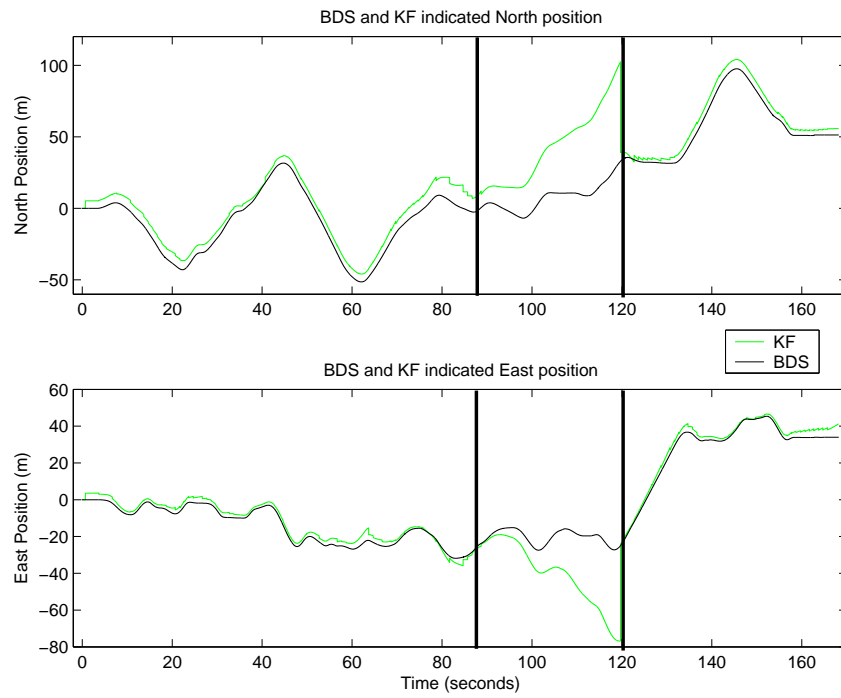


Figure 4.29 North and East Filter Performance, GPS Outage

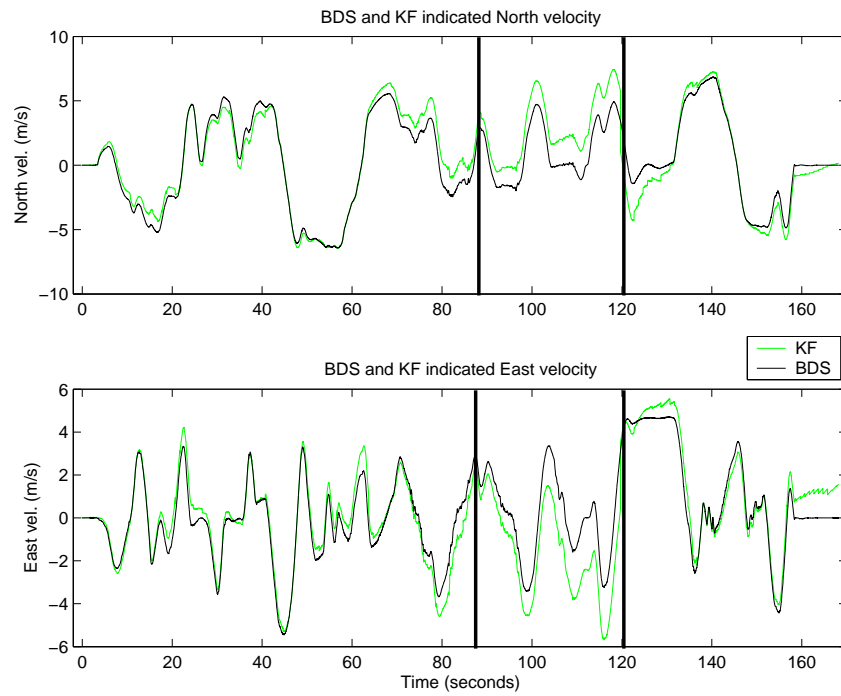


Figure 4.30 North and East Velocity Filter Performance, GPS Outage

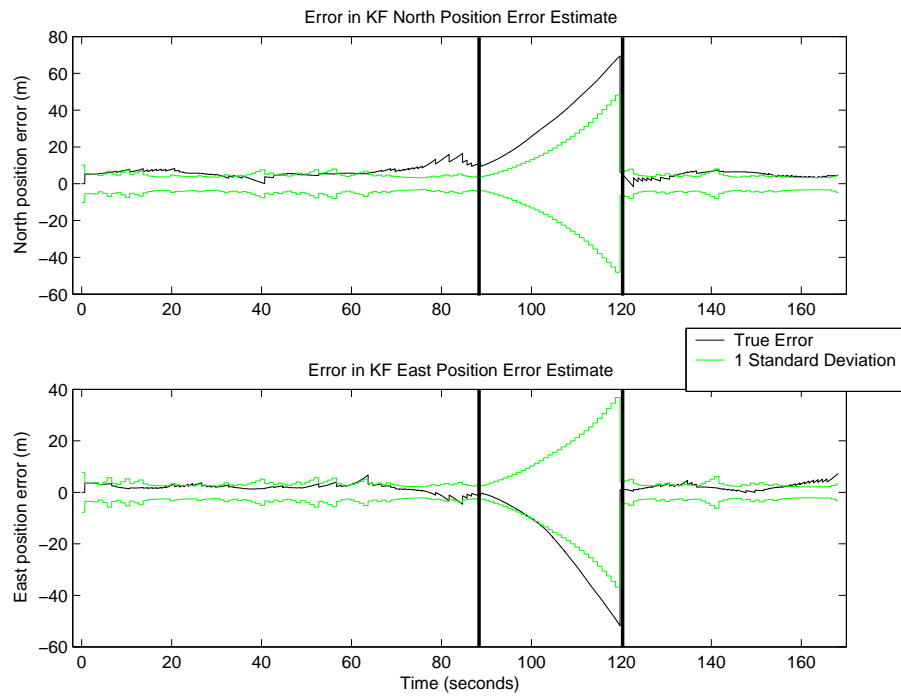


Figure 4.31 North and East Errors and Standard Deviations, GPS Outage

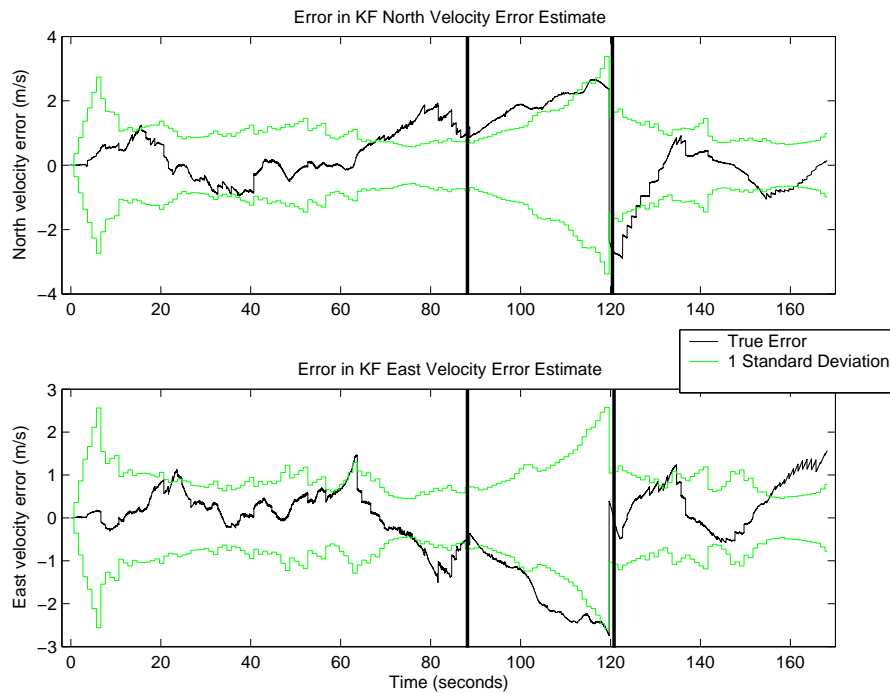


Figure 4.32 North and East Velocity Errors and Standard Deviations, GPS Outage

Table 4.6 GPS Outage Performance Errors

Filter Output	Mean Error	Error Std. Dev.	Pct. within 1σ
North Position	11.3 m	13.9 m	24.4
East Position	-2.10 m	10.9 m	71.4
Altitude	33.5 m	3.10 m	8.14
North Velocity	0.379 m/s	1.07 m/s	63.5
East Velocity	-0.154 m/s	0.944 m/s	64.3
Down Velocity	-0.0720 m/s	0.224 m/s	99.6
X-tilt	-0.000114 deg	0.592 deg	56.2
Y-tilt	0.127 deg	0.528 deg	59.8
Z-tilt	-3.26 deg	3.11 deg	76.5

When Table 4.6 and Table 4.4 (data set “short-term” without a GPS outage) are compared, it is apparent that the north and east direction and velocity error statistics are the only ones that changed significantly. The rest of the errors are essentially unchanged.

During the outage, the filter’s north and east position errors grow quickly until they are greater than 50 meters. Once GPS returns, the filter quickly corrects itself back to a good solution. The north and east velocity errors (Figure 4.30) also grow fairly quickly (compare with Figure 4.9 to see the differences). Again, once GPS returns, the filter’s velocity error quickly shrinks back down to a reasonable level. The error in these states is expected because GPS provides the filter with its most reliable position and velocity information. The filter raises its covariances for these states (Figures 4.31 and 4.32) when GPS is missing, showing the decreased confidence in its error estimates. Once GPS returns, the filter quickly lowers them to a more confident number. The results of this data set show that this system is viable in an environment where short GPS outages may occur. For an outage 30 seconds long,

the filter's position error grows to 50 or more meters, and over a longer period of time this error would only continue to grow.

4.6 Residuals

A sample of the filter's residuals from the "short-term" data set are shown in Figure 4.33. With a perfect model and perfectly tuned filter, the residuals would look like white Gaussian noise with a mean of zero. However, obvious patterns exist in these residuals, suggesting some sort of correlation with the golf cart's motion or an IMU error, and revealing a weakness in the system model. One way to correct this problem would be to improve/tweak the tuning of the filter and the content of the systems dynamics model, as long as too much complexity was not added. Another explanation for the patterns could be correlated errors in the GPS pseudorange measurements. During the last ten or fifteen seconds of the data set, the golf cart was sitting still, suggesting that a large part of the visible patterns is due to these unmodeled correlated errors in the GPS measurements.

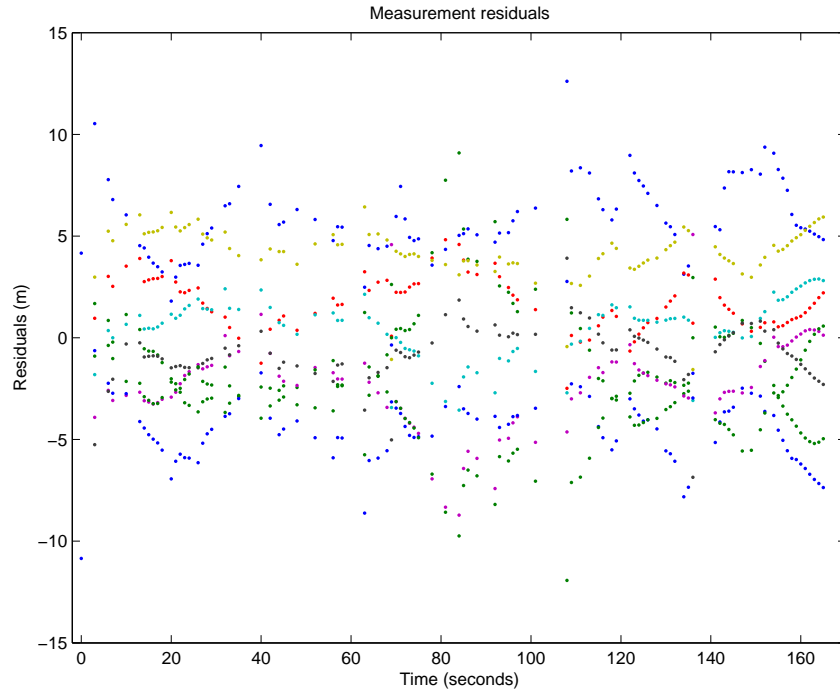


Figure 4.33 Filter Residuals, “short-term”

4.7 Summary

This chapter first described the methods used to collect and validate the IMU and GPS data. Next, the data processing was discussed and the filter’s alignment process, bias feedback and INS resets were explained. The filter’s noise and initial covariance values were listed and explained to set the background for the performance results. The short-term filter performance showed that the filter was capable of short-term accuracy, and the long-term data set validated the filter’s performance over a much longer period of time. The filter was further tested with the introduction of a short GPS outage, with decent, predictable results. Finally, a sample of filter residuals was shown not to be white as a perfect filter’s residuals would be.

V. Conclusions and Recommendations

5.1 Conclusions

The main conclusion that can be drawn from this research is that MEMS inertial instruments now exist that are accurate enough to be useful in a practical, comparatively low-cost INS/GPS integration. This conclusion has some caveats:

- The MT-9 has great utility for providing attitude. Its pitch and roll outputs are reasonably accurate, usually within 1 or 2 degrees, but its heading output tends to have much more error (up to 35-40 degrees a minute). Its ability to track heading is compromised by its unpredictable bias and high noise (especially in the gyroscopes), which prevent it from sensing earth's rotation rate. See the next section for recommendations about improving its heading performance.
- The MT-9 is not capable of performing stand-alone for more than a minute or so. After 60 seconds, the position error can be off by hundreds of meters, and it grows so quickly after this point that the solution quickly becomes useless. This limits its usefulness as a backup solution in case of a GPS signal outage. However, even a 30-second window can still be useful in case of signal blockages by buildings or trees, providing a reasonable solution until the GPS receiver regains signal lock.
- A weakness of this Kalman filter integration is that the filter is good at correcting the heading error *only* when the system is moving. Even if the system remains still for less than a minute, the overall heading error still grows significantly (see Figure 4.18). When in motion, the filter is quite good at correcting any existing error and then keeping the heading error less than five degrees or so.

A final conclusion is that the use of a tightly-coupled integration was a good choice for this system. The filter can use any number of satellites, not just a minimum of four. This somewhat mitigates the MT-9's poor stand-alone performance, allowing the filter still to use GPS measurements when a loosely-coupled system would be totally relying on the MT-9. A tightly-coupled system also helps determine INS error more efficiently because it uses data that has not been pre-processed as much.

5.2 *Recommendations*

This system performed well enough that further research and improvements would be useful. The following recommendations are provided as starting points for future work.

- Conduct more filter tuning. In its current incarnation, the filter's static heading performance leave much to be desired. As mentioned in Section 4.5.2.1, it appears that the relationship between the angular state noises and the gyroscope bias state noises is at least slightly wrong. If the right balance between these two noises is found, perhaps the static heading drift could be reduced.
- Test the system more thoroughly, this time using data time-stamped in real-time (as an actual system would use). It would also be an improvement to use GPS data provided by a separate receiver from the BDS, in order to remove any correlation with the truth data. More dynamic data would also be useful, such as that provided by a car or a small aircraft (which would be ideal because of the wide attitude variations during flight).
- Use all of the instruments available on the MT-9, specifically the magnetometers, which were only used in the alignment process in this research. There were two reasons the magnetometers were not used in this system. Firstly, this system development was originally intended to be used with any IMU, and in general, IMUs designed specifically for navigation do not have magnetometers.

Secondly, the long-term goal of this research is a component that can be embedded in other systems (see Section 1.1.5 for a list of potential applications) to provide navigation outputs. These types of environments can potentially be high in magnetic variations caused by electrical equipment or large metal structures, which would disrupt the readings of the magnetometers. Even with these potential disruptions, however, the magnetometer readings would be useful because any data, no matter how noisy, can be used by a Kalman filter [20]. The measurement model and measurement noise value can be tuned to allow some useful information to be gleaned from even the noisiest measurements. The most obvious benefit from the magnetometer measurements would address the weakest area of performance for the filter, which is the heading performance during static periods. The magnetometer readings would most likely greatly reduce the heading drift during these static periods, because the magnetic field would be unchanging, telling the filter that the perceived heading changes of the INS are actually due to the gyroscope drift.

- Add GPS pseudorange rate as a filter measurement. This would provide another velocity input to the system, and could also potentially improve its heading performance (see Section 4.5.2 for the rationale).
- Improve the alignment technique. One way to do this would be to perform the current static align, but then to perform some maneuvers that would allow the filter to figure out the heading error (the filter was effective at determining heading error when the system was in motion). The INS could then be reset using the filter's heading error estimate. Combining a more accurate alignment with the magnetometer data has the potential to improve greatly the filter's heading performance.
- Investigate the usefulness of simplifying of the Kalman filter's dynamics model. The large errors introduced by the MT-9's biased, noisy instruments may be so dominant that certain elements of the dynamics model could be simplified

without adverse effects. This would potentially allow smaller processors to run the code, which would use less power and be more practical for actual real-time implementation. Another benefit to a smaller model would be to allow the filter to update at a faster rate than 1 Hz, potentially improving its accuracy.

- Conduct more experiments to characterize the error of the MT-9. In this research only one gyroscope error was modeled. If more IMU data were to be collected and analyzed, different error characteristics might emerge, allowing for a more accurate system dynamics model.

Appendix A. Position Track of Relevant Data Sets

This appendix contains the position and altitude tracks, velocity data, and attitude data as recorded by the truth reference system (the NovAtel BDS).

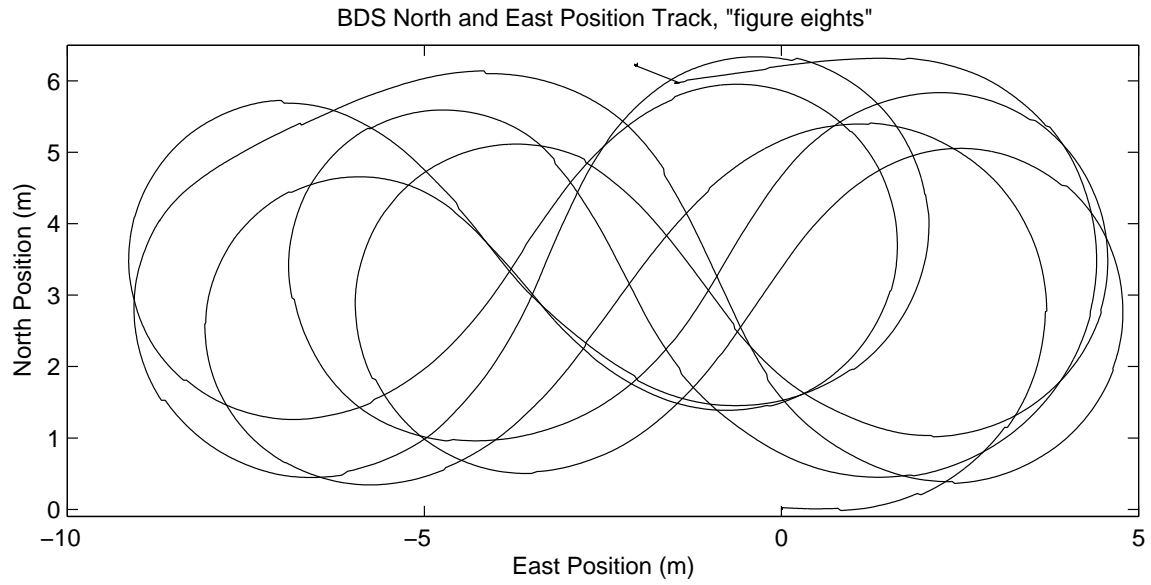


Figure A.1 BDS North and East Position Track, “figure eights”

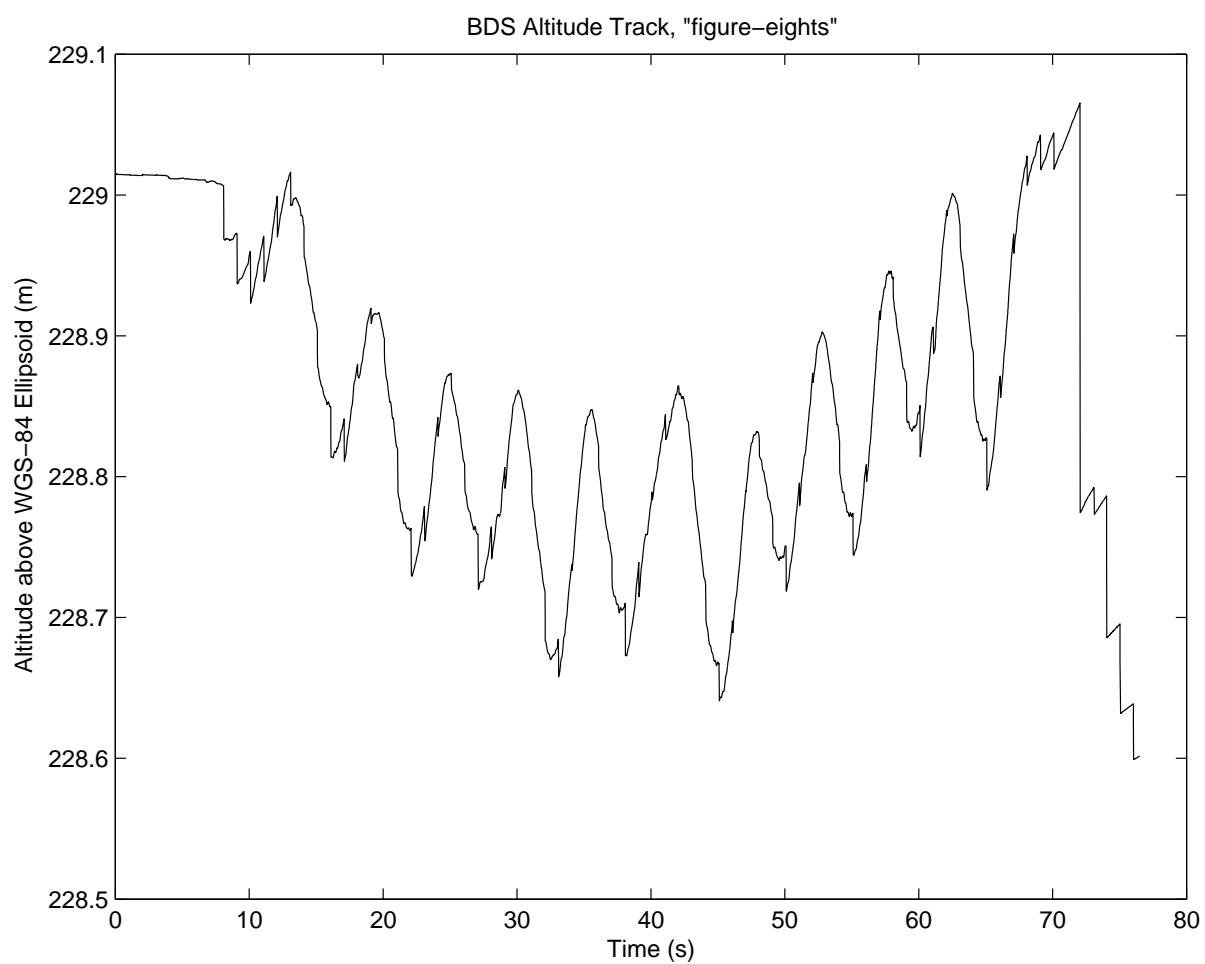


Figure A.2 BDS Altitude Track, "figure eights"

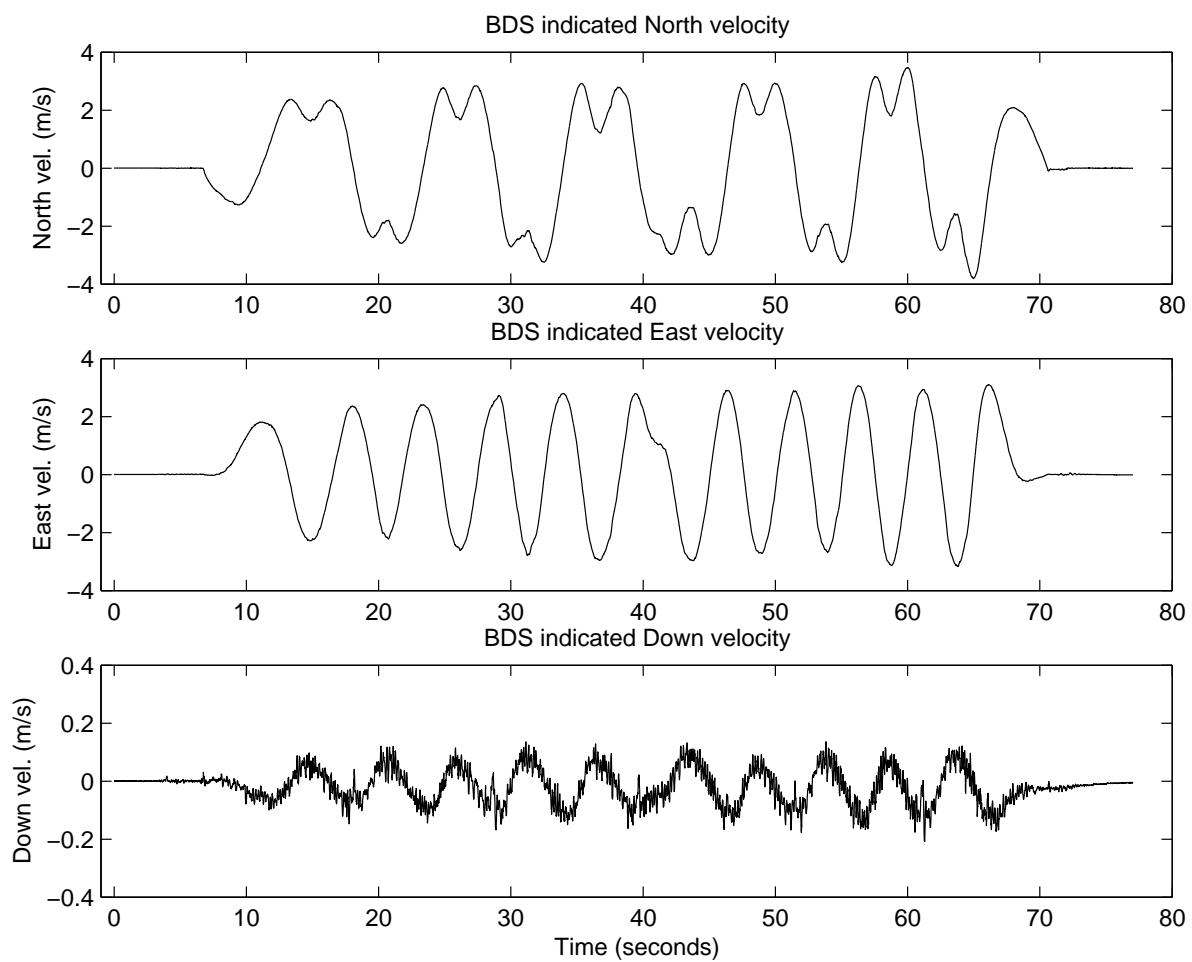


Figure A.3 BDS North, East, and Down Velocities, “figure eights”

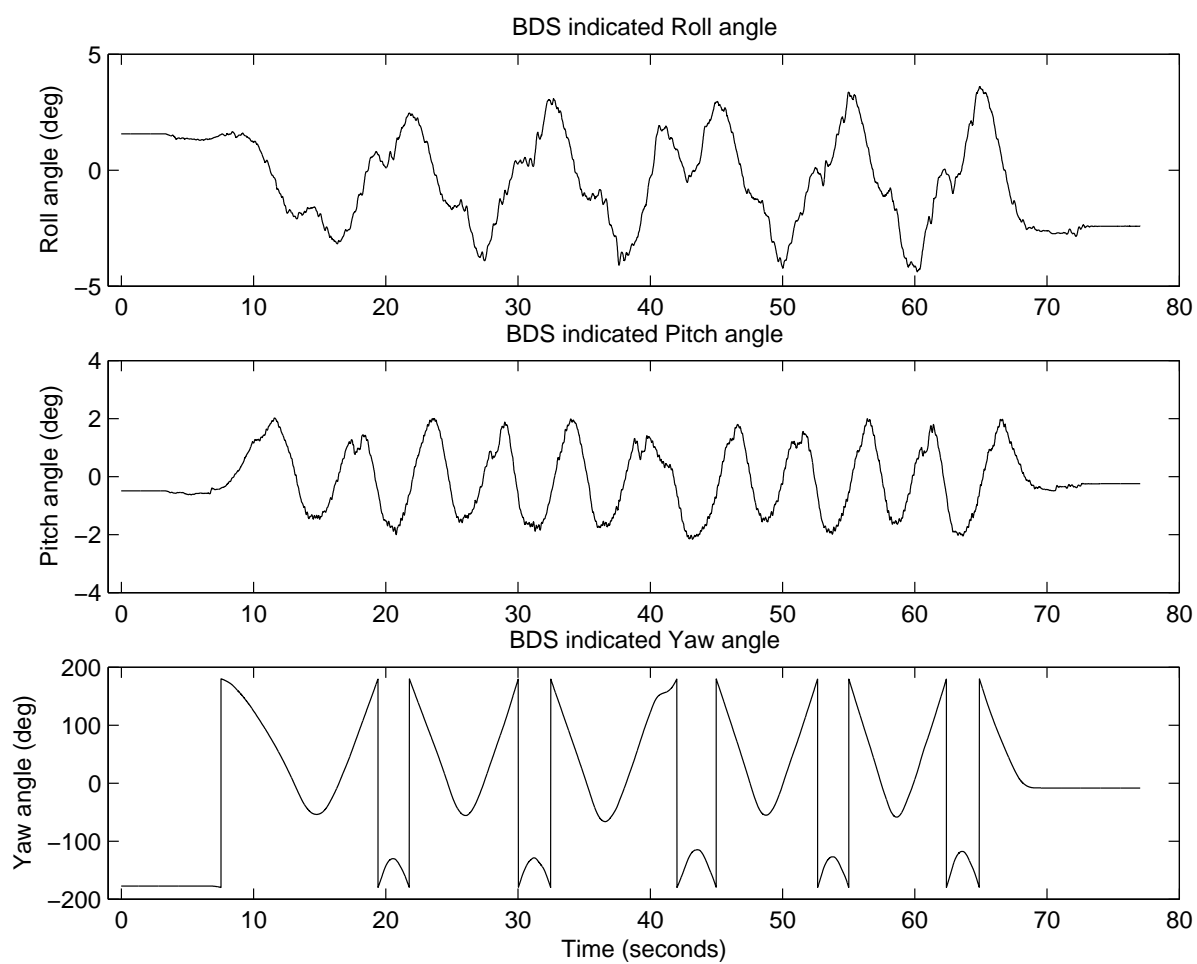


Figure A.4 BDS Roll, Pitch, and Yaw, “figure eights”

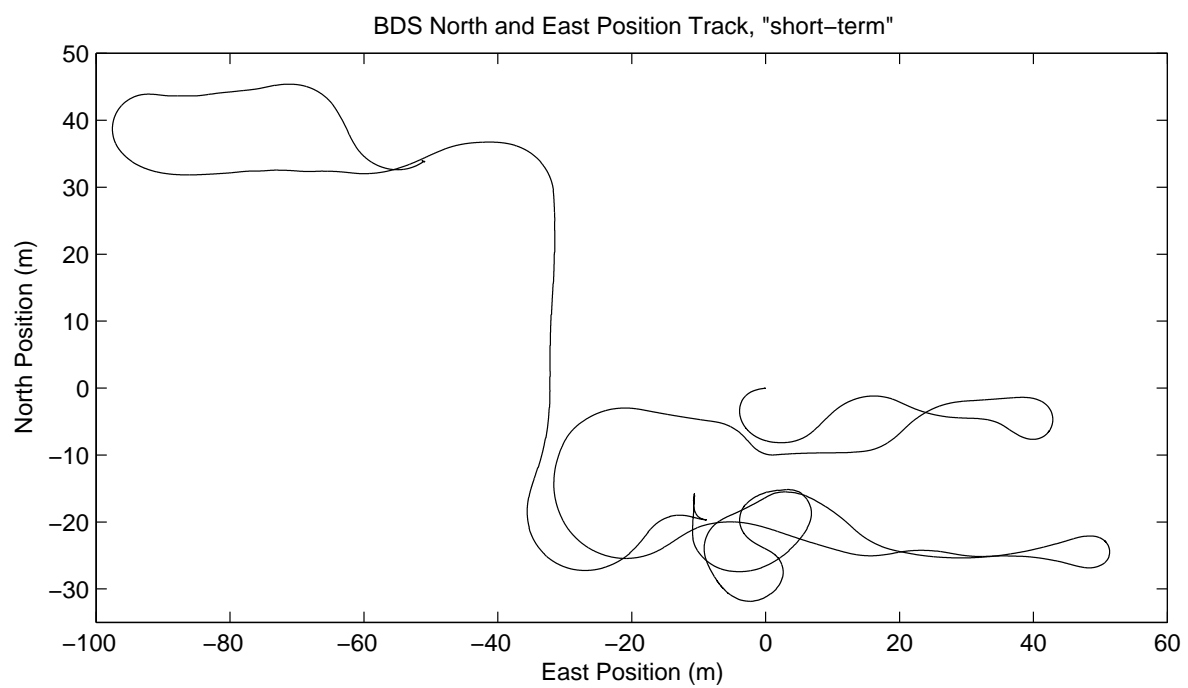


Figure A.5 BDS North and East Position Track, “short-term”

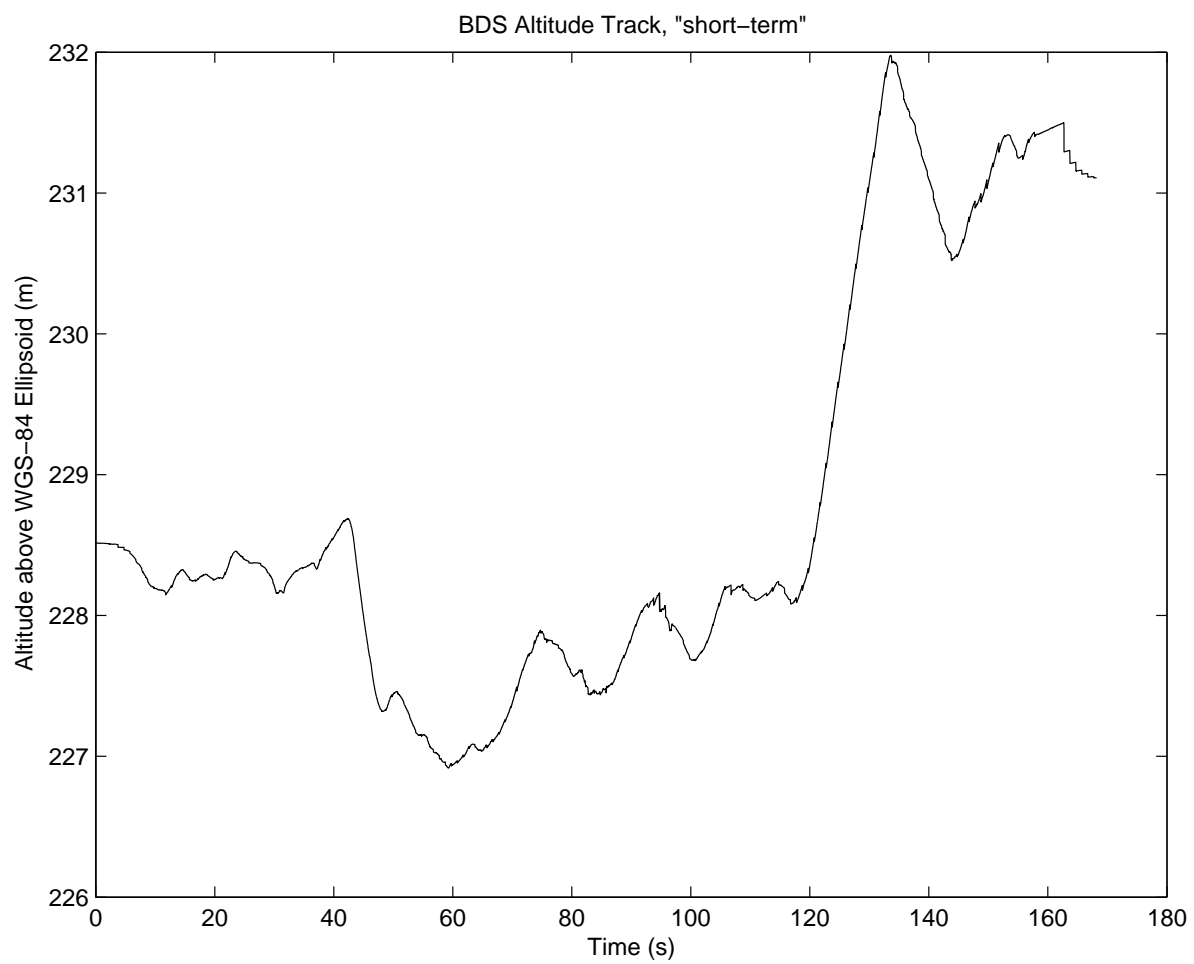


Figure A.6 BDS Altitude Track, "short-term"

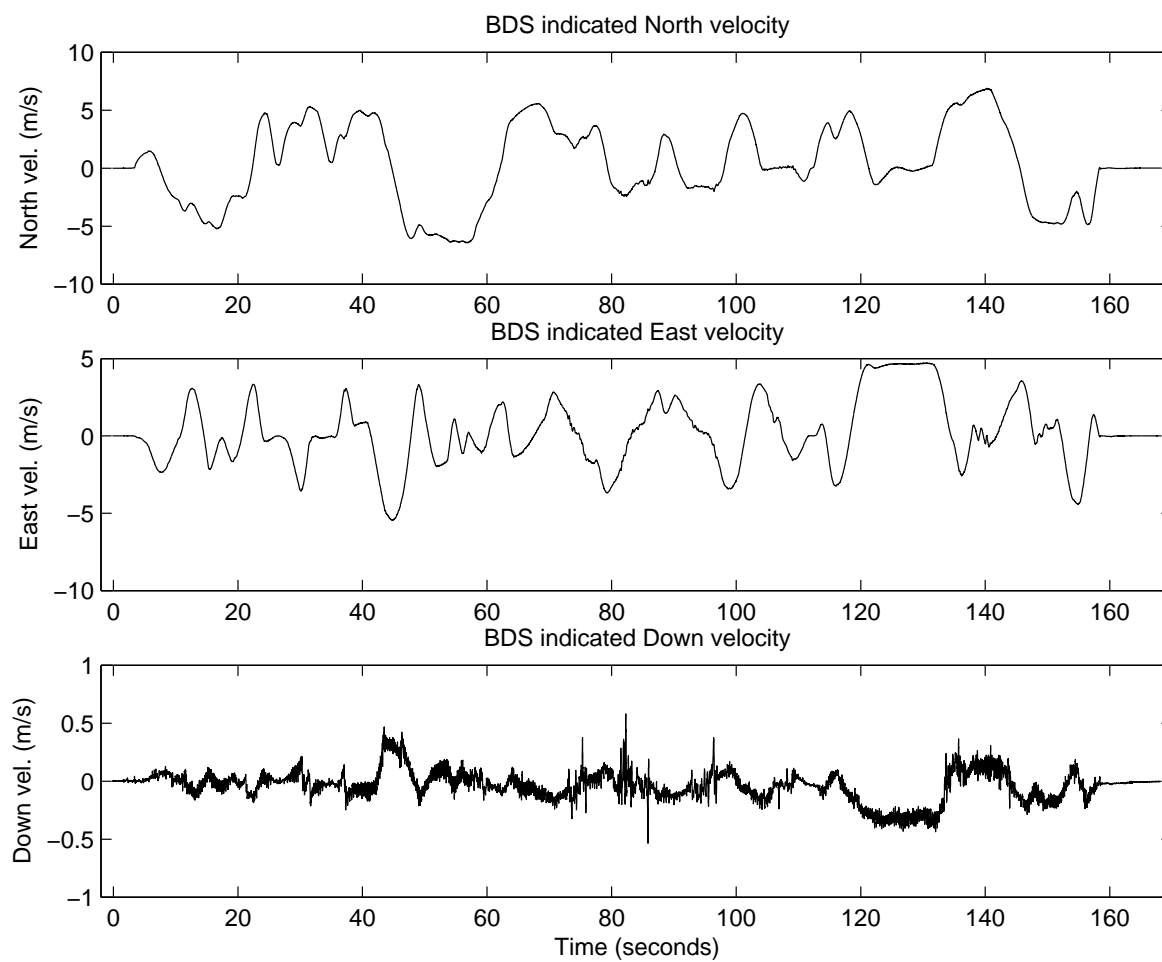


Figure A.7 BDS North, East, and Down Velocities, “short-term”

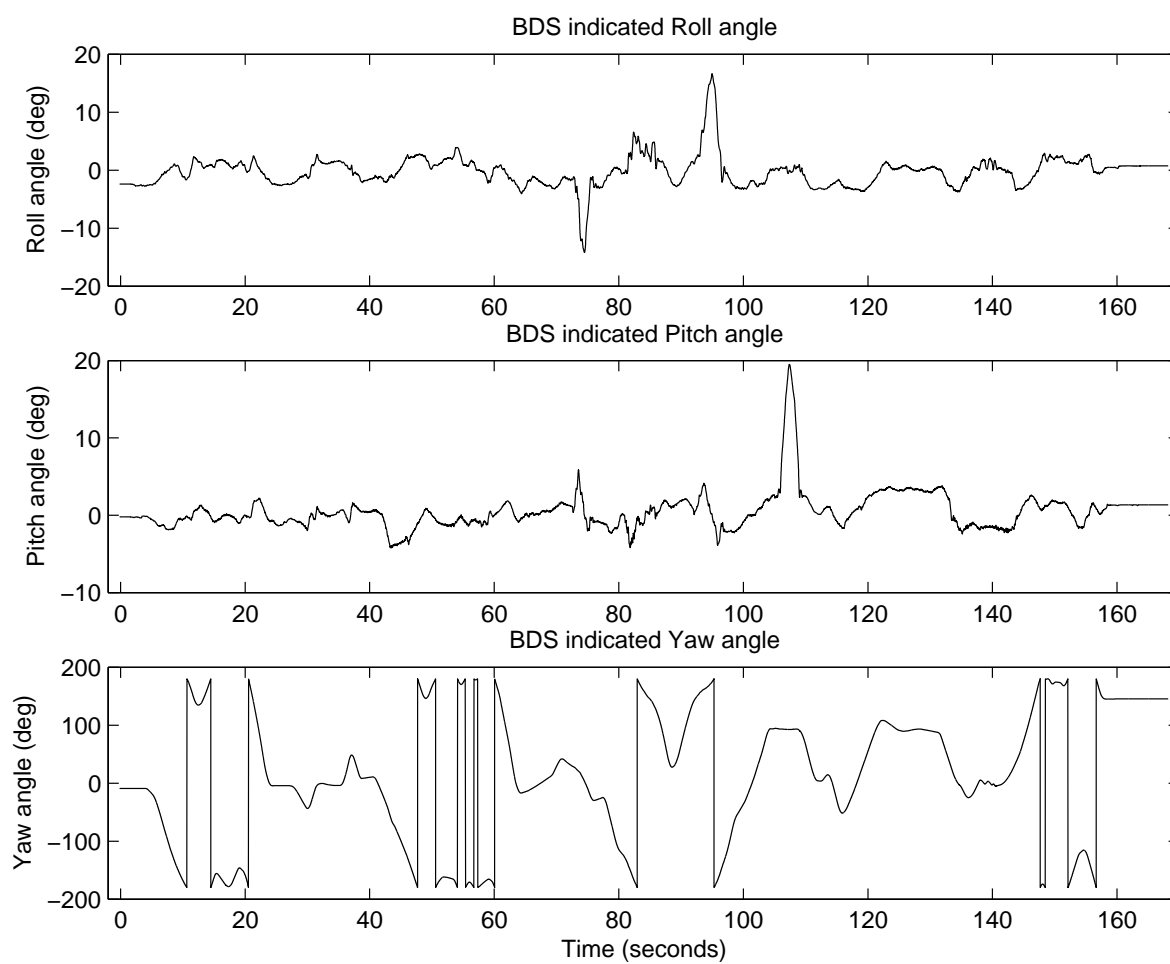


Figure A.8 BDS Roll, Pitch, and Yaw, “short-term”

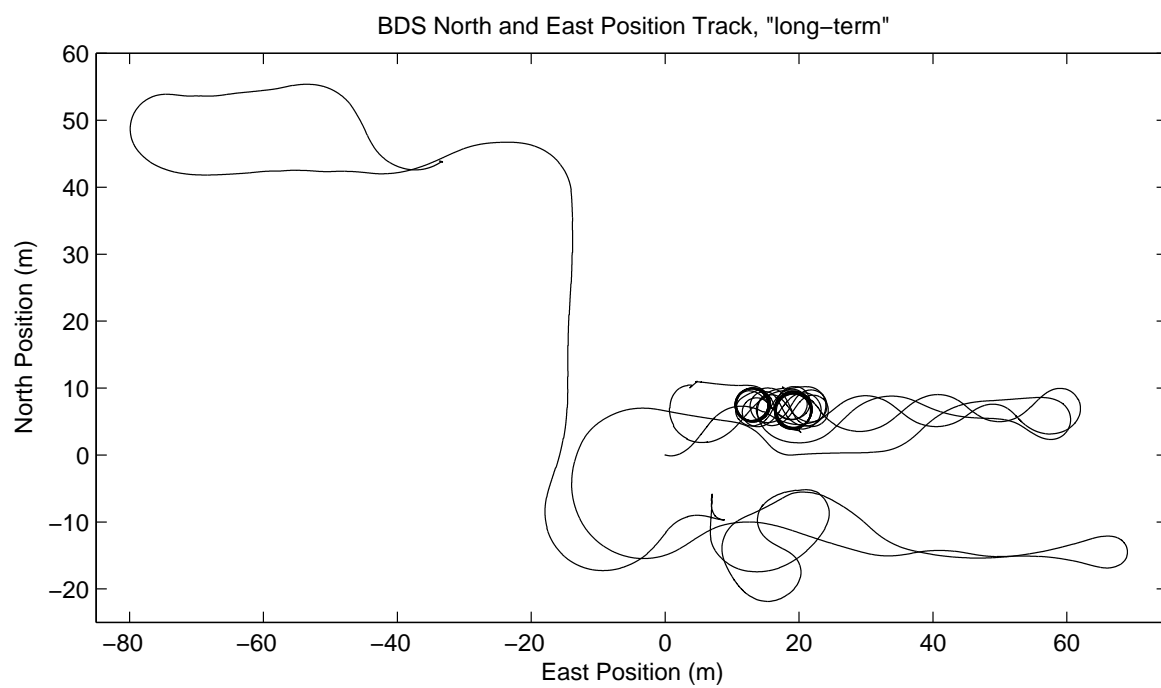


Figure A.9 BDS North and East Position Track, “long-term”

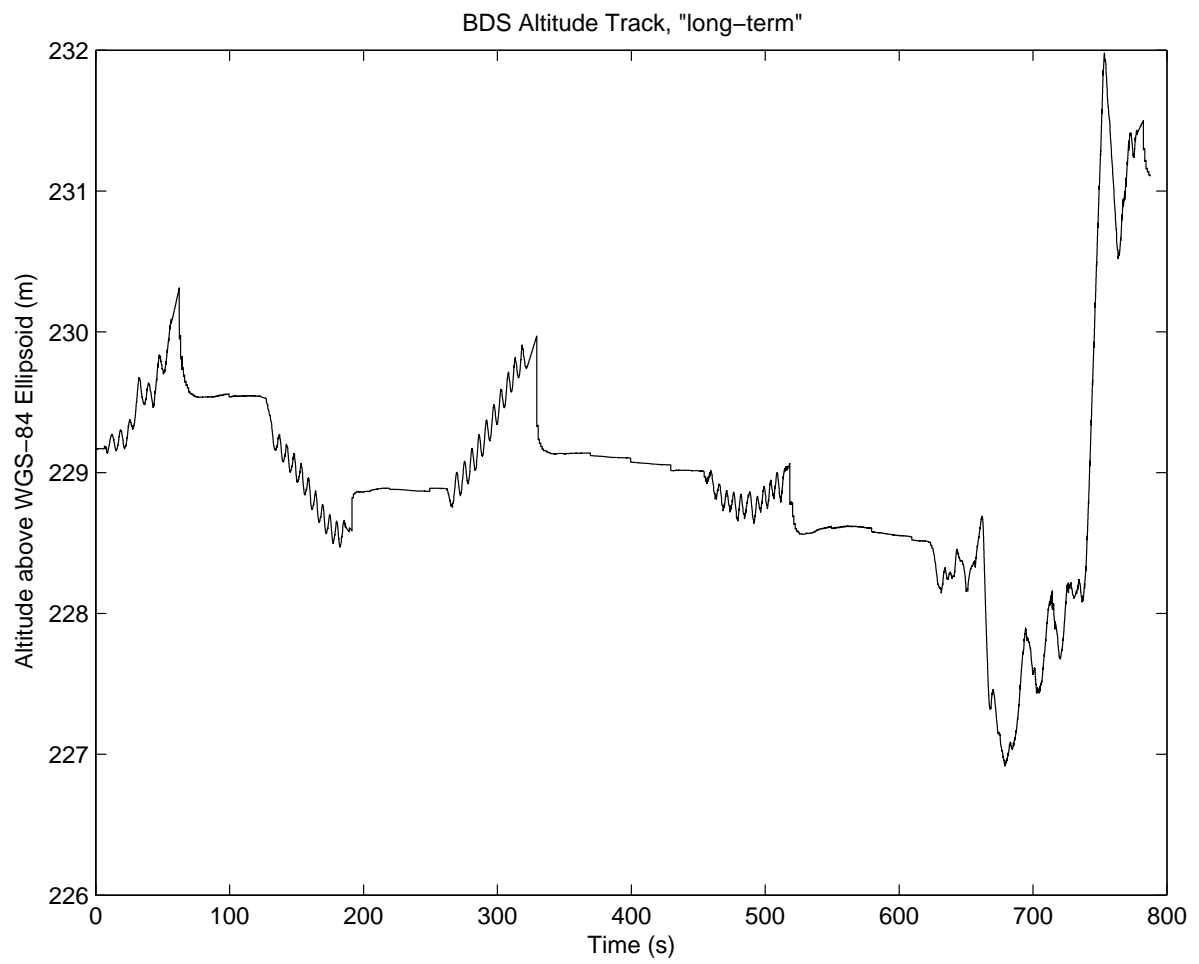


Figure A.10 BDS Altitude Track, "long-term"

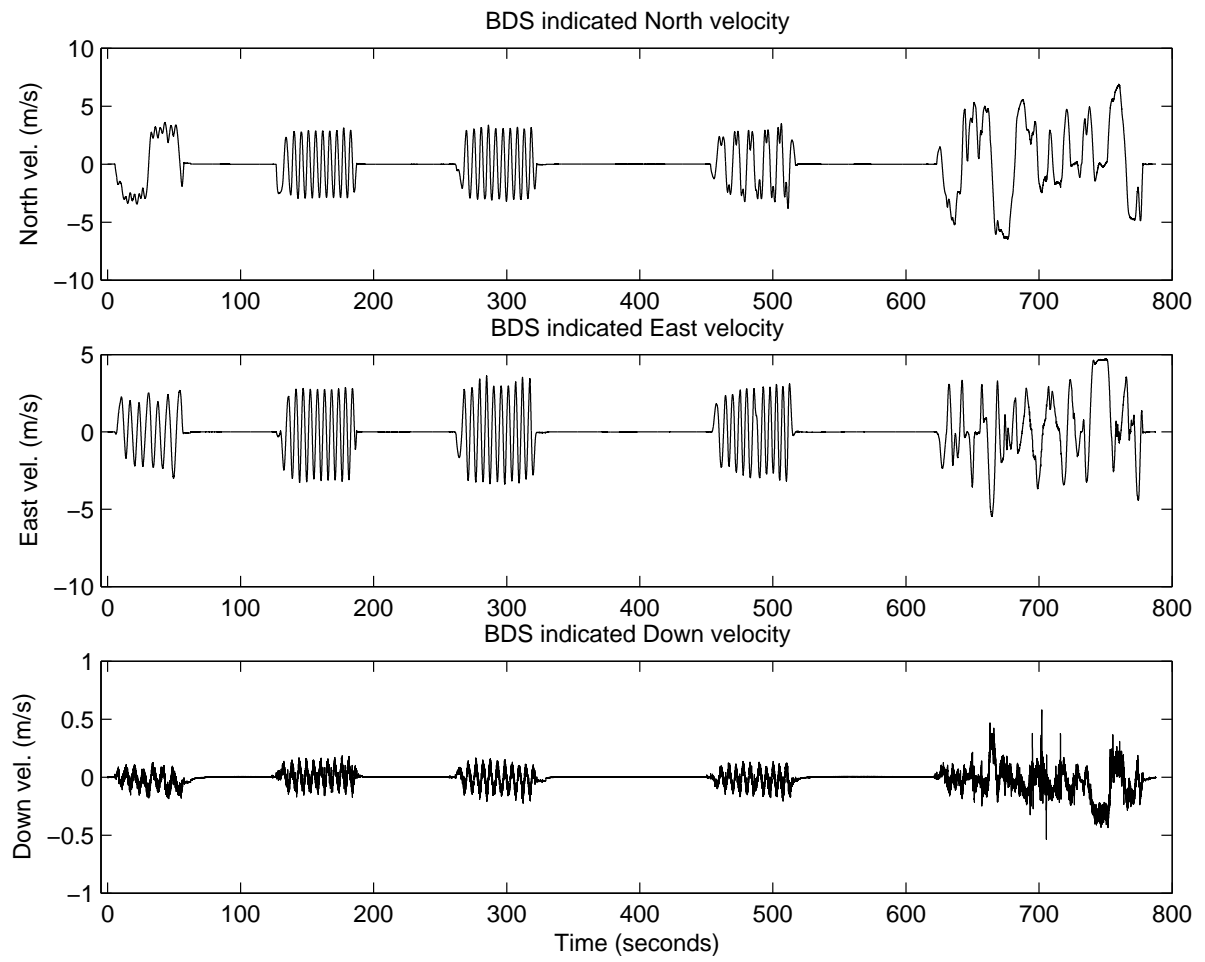


Figure A.11 BDS North, East, and Down Velocities, “long-term”

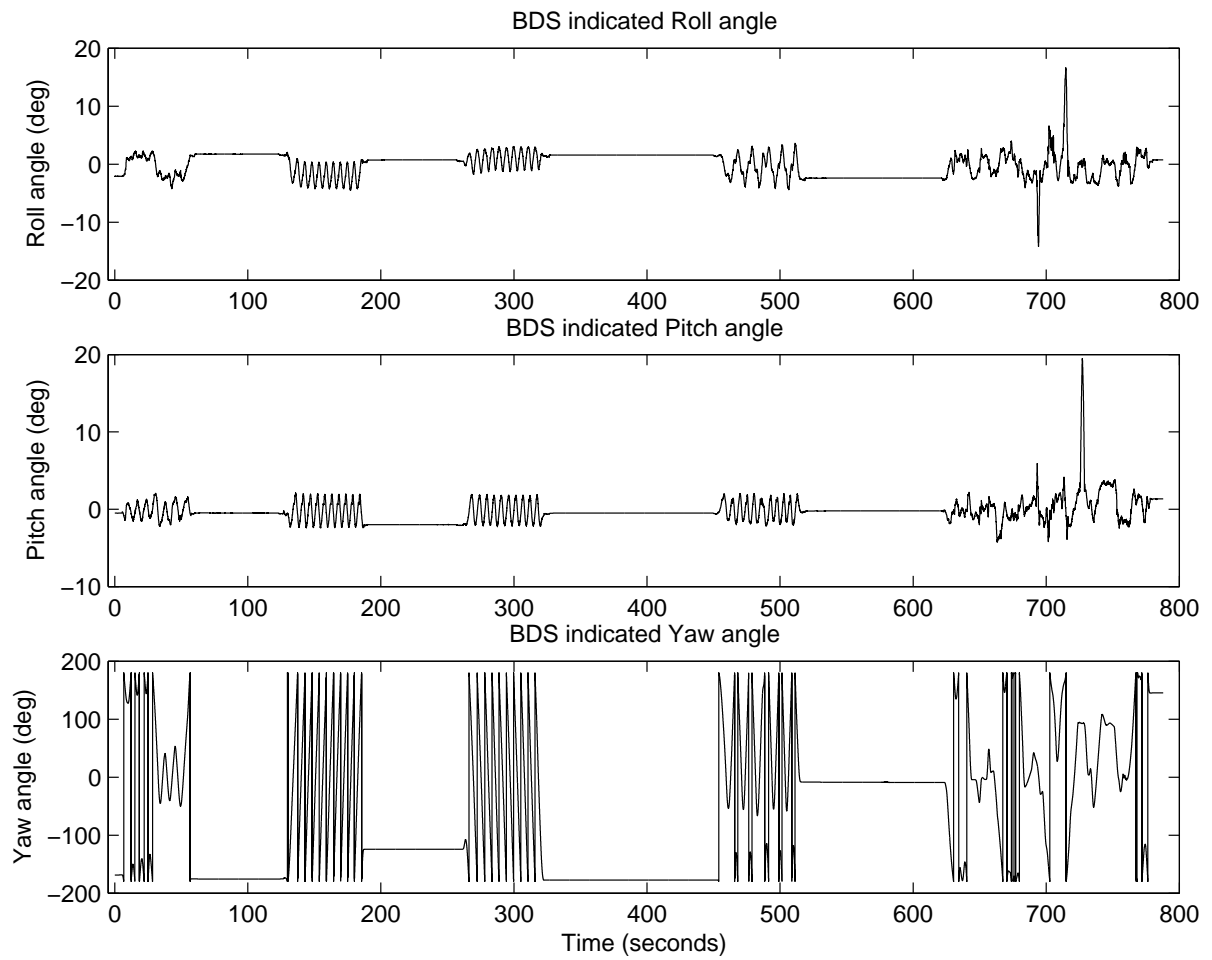


Figure A.12 BDS Roll, Pitch, and Yaw, “long-term”

Bibliography

1. <http://www.pangolin.co.nz/almanac/magvar.php>. Magnetic variance generated by GEOMAG (provided by U.S. National Geophysical Data Center). Accessed June 2004.
2. Acar, Cenk and Andrei M. Shkel. "Experimental evaluation and comparative analysis of commercial variable-capacitance MEMS accelerometers," *Journal of Micromechanics and Microengineering*, 634–645 (2003).
3. Anderson, Richard S., et al. "Evolution of Low-Cost MEMS Inertial System Technologies." *Proceedings of the ION GPS 2001*. 1332–1342. September 2001.
4. Balasz, Brian, et al. "GPS-Based 3D Navigation Visualization for High Altitude High Open Parachuting." *Proceedings of the ION 59th Annual Meeting/22nd CIGTF Guidance Test Symposium*. June 2003.
5. Bernstein, J., et al. "A Micromachined Comb-Drive Tuning Fork Rate Gyroscope." *Proceedings of IEEE MEMS '93*. 143–148. February 1993.
6. Bouska, Terry J. *Development and Simulation of a Pseudolite-based Flight Reference System*. MS thesis, Air Force Institute of Technology, 2003.
7. Brown, Alison and Dan Sullivan. "Precision Kinematic Alignment Using a Low-Cost GPS/INS System." *Proceedings of the ION GPS 2002*. 1094–1099. September 2002.
8. Chen, Wu, et al. "Attitude Determination with a Low Cost GPS/IMU Unit." *Proceedings of the ION 58th Annual Meeting/CIGTF 21st Guidance Test Symposium*. 465–472. June 2002.
9. Collin, Jussi, et al. "Unaided MEMS-Based INS Application in a Vehicular Environment." *Proceedings of the ION GPS 2001*. 1343–1348. September 2001.
10. Department of Defense. *Department of Defense World Geodetic System 1984: its definition and relationship with local geodetic systems*. DMA TR 8350.2. September 1987.
11. El-Diasty, Mohammed and Ahmed El-Rabbany. "Adaptive Noise Reduction Model for MEMS-Based Inertial Sensors." *Proceedings of the ION NTM 2003*. 636–640. January 2003.
12. Ford, Tom, et al. "MEMS Inertial on an RTK GPS Receiver: Integration Options and Test Results." *Proceedings of the ION NTM 2003*. 466–477. January 2003.
13. Goad, C. C. and L. Goodman. "A Modified Hopfield Tropospheric Refraction Correction Model." *Proceedings of the Fall Annual Meeting of the American Geophysical Union*. 1974.

14. Gutiérrez, José R. *Multipath and GPS Signal Jamming Mitigation through Multiple Model Adaptive Estimation Applied to Ultra-Tightly Coupled GPS/INS Architecture*. MS thesis, Air Force Institute of Technology, 2003.
15. Han, Ki-Ho and Young-Ho Cho. "Self-Balanced Navigation-Grade Capacitive Microaccelerometers Using Branched Finger Electrodes and Their Performance for Varying Sense Voltage and Pressure," *Journal of Microelectromechanical Systems*, 12(1):11–20 (February 2003).
16. Henderson, Paul E. *Development and Testing of a Multiple Filter Approach for Precise DGPS Positioning and Carrier-Phase Ambiguity Resolution*. MS thesis, Air Force Institute of Technology, 2001.
17. Hide, Christopher, et al. "Adaptive Kalman Filtering for Low Cost INS/GPS." *Proceedings of the ION GPS 2002*. 1143–1147. September 2002.
18. Klobuchar, J. A. "Ionospheric Effects on GPS." *Global Positioning System: Theory and Applications 1*. Progress in Astronautics and Aeronautics, edited by Bradford W. Parkinson and James J. Spilker Jr., Washington, DC: American Institute of Aeronautics and Astronautics, Inc., 1996.
19. Martin, Dr. Michael K., et al. "Benefits of Low-Cost INS/GPS to Augment Land Navigation." *Proceedings of the ION GPS 1999*. 1121–1132. September 1999.
20. Maybeck, Peter S. *Stochastic Models, Estimation, and Control, Volume 1*. Mathematics in Science and Engineering, Vol. 141-1, Arlington, Virginia: Navtech Book and Software Store, 1994.
21. M'Closkey, Robert T., et al. "System Identification of a MEMS Gyroscope," *Journal of Dynamic Systems, Measurement, and Control*, 123:201–210 (June 2001).
22. MicroStrain, Inc. *3DM-G User Manual*. Technical Report. Williston, Vermont, 2003.
23. Misra, Pratap and Per Enge. *Global Positioning System Signals, Measurements, and Performance*. Lincoln, Massachusetts: Ganga-Jamuna Press, 2001.
24. Mueller, Conrad E. and Joel G. Hanse. "Honeywell MEMS Inertial Systems: Requirements and Development." *Proceedings of the ION 55th Annual Meeting*. 451–461. June 1999.
25. Parkinson, Bradford W. "GPS Error Analysis." *Global Positioning System: Theory and Applications 1*. Progress in Astronautics and Aeronautics, edited by Bradford W. Parkinson and James J. Spilker Jr., Washington, DC: American Institute of Aeronautics and Astronautics, Inc., 1996.

26. Pervan, Boris, et al. "Performance Analysis of Carrier-Phase DGPS Navigation for Shipboard Landing of Aircraft," *NAVIGATION: Journal of the Institute of Navigation*, 50(3):181–191 (Fall 2003).
27. Raquet, John F. MATLAB code distributed in EENG 735: Integrated Navigation Systems, Summer Quarter 2003.
28. Raquet, John F. and David L. M. Warren. "Broadcast vs. Precise GPS Ephemerides: a Historical Perspective," *GPS Solutions*, 7(3):151–156 (December 2003).
29. Rios, Jose A. and Elecia White. "Fusion Filter Algorithm Enhancements For a MEMS GPS/IMU." *Proceedings of the ION GPS 2001*. 1382–1393. September 2001.
30. Rogers, Robert M. *Applied Mathematics in Integrated Navigation Systems*. AIAA Education Series, Reston, Virginia: American Institute of Aeronautics and Astronautics, Inc., 2000.
31. Snodgrass, Britt and John Raquet. "The CIGTF High Accuracy Post-Processing Reference System (CHAPS)." *Proceedings of the ION GPS 1994*. 755–763. September 1994.
32. Spilker Jr., J. J. "Tropospheric Effects on GPS." *Global Positioning System: Theory and Applications 1*. Progress in Astronautics and Aeronautics, edited by Bradford W. Parkinson and James J. Spilker Jr., Washington, DC: American Institute of Aeronautics and Astronautics, Inc., 1996.
33. Titterton, David H. and John L. Weston. *Strapdown Inertial Navigation Technology*. IEE Radar, Sonar, Navigation and Avionics Series 5, Stevenage, Herts., UK: Peter Peregrinus Ltd., 1997.
34. Tredway, Reece, et al. "Using the GPS to Collect Trajectory Data for Ejection Seat Design, Validation, and Testing." *Proceedings of the ION GPS 2001*. September 2001.
35. Vallot, Lawrence, et al. "Design and Flight Test of a Differential GPS/Inertial Navigation System for Approach/Landing Guidance," *NAVIGATION: Journal of the Institute of Navigation*, 38(2):321–340 (Summer 1991).
36. van Graas, Frank and James L. Farrell. "GPS/INS - A Very Different Way." *Proceedings of the ION 57th Annual Meeting/CIGTF Biennial Guidance Test Symposium*. 715–721. June 2001.
37. Wagner, Jorg F. and Gunther Kasties. "Improving the GPS/INS Integrated System Performance by Increasing the Distance Between GPS Antennas and Inertial Sensors." *Proceedings of the ION NTM 2002*. 103–115. January 2002.

38. White, Elecia and Jose A. Rios. “FAA Certification of a MEMS Attitude and Heading Reference System.” *Proceedings of the ION GPS 2001*. 1349–1360. September 2001.
39. XSens Technologies B.V. *MT-9B Data Sheet*. Technical Report. Enschede, The Netherlands, 2003.

Vita

Jonathan M. Neu grew up in Logansport, Indiana, where he was homeschooled for much of his education. He graduated from Logansport High School in 1996 and then attended Cedarville University in Cedarville, Ohio. In 2001, he graduated with a Bachelor of Science in Electrical Engineering.

Jonathan currently works for the Engineering Directorate of the Aeronautical Systems Center at Wright-Patterson AFB, Ohio, doing logistics studies using the Logistics Composite Model. He is married with one daughter.

Permanent address: ON FILE

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 074-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY) 14-09-2004		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From - To) June 2003 - September 2004	
4. TITLE AND SUBTITLE A TIGHTLY-COUPLED INS/GPS INTEGRATION USING A MEMS IMU				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Neu, Jonathan M.				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 640 WPAFB OH 45433-8865				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GE/ENG/04-19	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFRL/HEPA John A. Plaga 2800 Q Street WPAFB OH 45433 DSN: 785-1166 email: John.Plaga@wpafb.af.mil				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT <p>Micro-Electro-Mechanical Systems (MEMS) technology holds great promise for future navigation systems because of the reduced size and cost of MEMS inertial sensors relative to conventional devices. Current MEMS devices are much less accurate than standard inertial sensors, but they can still be useful. In this thesis, data was recorded from an inexpensive MEMS inertial measurement unit and integrated with GPS measurements using a tightly-coupled Kalman filter. The overall goal of this research is to investigate the usefulness of MEMS sensors for a small, real-time, low-cost INS/GPS integration.</p> <p>A golf cart was used to collect dynamic data, along with a commercial INS/GPS system to provide reference data. This data was then post-processed, and the filter's performance in the position, velocity, and attitude outputs were evaluated by comparing them to the reference system. The important system features of system alignment, bias feedback, and INS resets are described, and the filter's performance is analyzed using its estimate and covariance outputs and comparing them to the true error. Filter residuals are also shown and discussed.</p> <p>The final results show that, with adequate processing available, the INS/GPS filter using the MEMS instruments provides good position, velocity, and attitude results over a period of up to 15 minutes, as long as the data is at least somewhat dynamic. Without vehicle motion, the vehicle yaw state tends to wander excessively, due to the bias and noise of the MEMS gyroscopes. Over a long static period, the filter's position outputs would most likely diverge and become unstable. Recommendations are made to combat this problem, among them to conduct more characterization of the MEMS sensors, and to add GPS velocity measurements as an input to the filter.</p>					
15. SUBJECT TERMS GPS, INS, INS/GPS, integration, MEMS, IMU, Kalman filter, tightly coupled					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 112	19a. NAME OF RESPONSIBLE PERSON Dr. John F. Raquet, AFIT/ENG
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code) (937) 255-3636, ext 4580 (John.Raquet@afit.edu)